ISMAEL LUIZ HARTMANN CERDEIRA GUMIEL

SISTEMA ESPECIALISTA MONITOR DE REDES DE COMPUTADORES

FLORIANÓPOLIS - SC
2003

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ismael Luiz Hartmann Cerdeira Gumiel

SISTEMA ESPECIALISTA MONITOR DE REDES DE COMPUTADORES

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Vitório Bruno Mazzola

Florianópolis, março de 2003

SISTEMA ESPECIALISTA MONITOR DE REDES DE COMPUTADORES

Ismael Luiz Hartmann Cerdeira Gumiel

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação na Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____
Professor Vitório Bruno Mazzola, Dr.

Banca Examinadora

_____
Professor Vitório Bruno Mazzola, Dr. (Orientador)

_____
Professor Cláudio César de Sá, Dr. (Co-orientador)

_____
Professor xxxxxxxxxxxxxxxxxxxxxxxxxx, Dr.

_____
Professor xxxxxxxxxxxxxxxxxxxxxxxxxx, Dr.

"Atribuir às próprias idéias um valor
absoluto e definitivo é tentar matar,
pelo silêncio, parte importante de si mesmo".
Artur da Távola

Agradeço ao meu orientador Prof. Vitório
e ao meu co-orientador Prof. Claudio,
que suportaram pacientemente e prontamente a todas
as minhas dúvidas e questionamentos.
Em especial a minha família que soube entender
e principalmente apoiar este momento da minha vida.

ÍNDICE

RESUMO ..... **IX**
ABSTRACT ..... **X**
LISTA DE FIGURAS e TABELAS ..... **XI**
LISTA DE SIGLAS ..... **XII**
CAPÍTULO 1 – Introdução **14**
1.1. Cenário **14**
1.2. Contexto do Problema **16**
1.3. Problema **17**
1.4. Motivação **18**
1.4.1. Justificativa **19**
1.5. Objetivos **19**
1.5.1. Objetivo Geral **20**
1.5.2. Objetivos Específicos **20**
1.6. Organização do Trabalho **21**
CAPÍTULO 2 –Gerenciamento de edes..... **22**
2.1. Contexto **22**
2.2. Gerenciamento de Falhas **23**
2.3. Gerenciamento de Contabilização **24**
2.4. Gerenciamento de Configuração **25**
2.5. Gerenciamento de Desempenho **26**
2.6. Gerenciamento de Segurança **27**
2.7. Sistema de Gerenciamento de Redes **27**
CAPÍTULO 3 – Protocolo SNMP **31**
3.1. O Modelo de Gerenciamento SNMP **32**
3.2. Definição dos Relacionamentos Administrativos **35**
3.3. Operações SNMP **37**
CAPÍTULO 4 – Monitoriramento Remto (RMON) **40**

4.1. Objetivos do Protocolo RMON **41**
4.2. Controle de Monitores Remotos **43**
4.3. RMON 1 e RMON 2 ..... **44**
4.4. Base de Dados RMON-MIB **46**
4.4.1. Grupos da RMON-MIB 1 ..... **46**
4.4.2. Grupos da RMON-MIB 2 ..... **47**
CAPITULO 5 Ferramentas Gerenciadoras **49**
5.1. Olho Vivo **49**
5.2. SAGRES **52**
5.3. Cisco Works **54**
5.4. Open View **55**
5.5. Tivoli Enterprise **56**
CAPITULO 6 Sistemas Especialistas **58**
6.1. Introdução **58**
6.2. Estrutura de Sistemas Especialistas **64**
6.2.1. Base de Conhecimento **65**
6.2.2. Regras de Produção **65**
6.2.3. Máquina de Inferência **66**
6.2.4. Interface com o Usuário **67**
6.3. Conhecimento Procedural x Conhecimento Declarativo **67**
6.4. Métodos de Extração de Conhecimento **68**
6.4.1. Encadeamento Progressivo (Forward Chaining) **69**
6.4.1.1. Resolução de Conflito **70**
6.4.2. Encadeamento Regressivo (Backward Chaining) **71**
6.4.3. Resumo **71**
CAPITULO 7 IMPLEMENTAÇÃO **72**
7.1. O Modelo **73**

7.2. Interface SNMP **75**
7.3. Variáveis Selecionadas **77**
7.4. Regras de Produção **79**
7.5. O Sistema Especialista Proposto **81**
7.6. Simulação **84**
CAPÍTULO 8 – CONCLUSÃO **91**
8.1. Resumo do Trabalho **91**
8.2. Principais Resultados **91**
8.3. Metodologia Adotada **92**
REFERÊNCIAS BIBLIOGRÁFICAS **94**
ANEXO **97**

RESUMO

O propósito deste trabalho, que abrange a área de informática, mais especificamente a gerência de redes e a Inteligência Artificial é implementar uma ferramenta para auxiliar a gerência de redes locais ou distribuídas.

Através da leitura de parâmetros da rede, como por exemplo fragmentação de pacotes ou nível de “broadcast”, e a comparação destes com valores limites pré-definidos, esta ferramenta indica ao administrador da rede possíveis problemas com os elementos da rede como roteadores, hub’s, switches etc, relacionados a parâmetros fora dos limites pré-estabelecidos.

Para o desenvolvimento foram utilizadas tecnologias como agentes RMON, software instalado em elementos de rede, que colhem e armazenam parâmetros de rede para posterior utilização por um software gerente, protocolo SNMP, protocolo que facilita a troca de informações de gerência entre agente e gerente, linguagem de programação C e linguagem Prolog.

Foi desenvolvida uma “SHELL” de sistema especialista, com a qual o especialista irá interagir, inserindo o seu próprio conhecimento, adequando a ferramenta às suas necessidades, de maneira que a base de conhecimento desta “Shell” crescerá até o limite do próprio conhecimento do administrador.

ABSTRACT

The purpose of this work that is related more specifically with the network management and the Artificial Intelligence topics, is to implement a tool to support the administration of a local or distributed networks.

By the reading of the network parameters, like fragmentation or broadcast level for example, and the comparison of this values with pre-seted values, the tool informs to the network administrator of the network, probabilities of problems with the network elements, like routers, switches etc.

Using RMON agents, software installed in some network elements that captures and store network parameters for the usage of another software called manager, SNMP protocol, that is a protocol who facilitates de communication between the manager and the agent, C and Prolog languages, a shell was developed.

The specialist will interact with this tool, adding his own knowledge, adapting the tool to his needs, so the knowledge base will increase until the limit of the proper specialist knowledge.

LISTA DE FIGURAS E TABELAS

Figura 1: Gerentes e Agentes em Rede Distribuída 29
Figura 2: Estrutura do Protocolo TCP/IP localizando o SNMP 31
Figura 3: Dinâmica do Protocolo SNMP 34
Figura 4: Formato das mensagens SNMP 38

Figura 5: Diferenças na Atuação da RMON1 e 2 45
Figura 6: Diagrama de Blocos do Olho Vivo 51
Figura 7: Diagrama de Blocos SAGRES 52
Figura 8: Estrutura Convencional de um Sistema Especialista 62
Figura 9: Diagrama de Contexto do Modelo Proposto 73
Figura 10: Diagrama de Blocos do Sistema 73
Figura 11: Contexto da Interface SNMP 75
Figura 12: Diagrama de Blocos Interface SNMP 76
Figura 13: Detalhe em blocos do sistema especialista 82
Figura 14: Detalhe do Menu do Sistema Especialista 83
Figura 15: Menu do programa Interface SNMP 84
Figura 16: Consulta valor de variável na interface SNMP 85
Figura 17: Inclusão de variável a ser monitorada 85
Figura 18: Remoção de variável da interface SNMP 86
Figura 19: Listagem de todas as variáveis e seus valores 86
Figura 20: Arquivo fatos_rmon.txt para simulação 87
Figura 21: Arquivo de regras (base de conhecimento) do Monitor 88
Figura 22: Dinâmica de funcionamento do monitor 88
Figura 23: Simulação do monitor 89
Tabela 1: Comparativo entre Programação Convencional e SE 59
Tabela 2: Exemplo Base de Conhecimento 69
Tabela 3: Descrição dos Ciclos de uma MI com Encadeamento Para-frente 70
Tabela 4: Variáveis do Sistema 78
Tabela 5: Regras do Sistema 79

LISTA DE ABREVIATURAS

AIX – Sistema operacional da IBM.

API – "Aplication Program Interface" uma biblioteca que dá acesso a várias funcionalidades do sistema operacional, ou outros acessos.

DNS – “Domain Name Server" ou em português "Servidor de Nomes de Domínio". Todo domínio registrado na Internet possui dois ou mais Servidores respondendo a seu nome 24h/dia, caso contrário o domínio não seria localizado na Internet.

FTP – "File Transfer Protocol" permite a transferência de arquivos entre dois computadores: o cliente e o servidor.

HTTP – “Hyper Text Transfer Protocol” (Protocolo de Transferência de Hipertexto) é um protocolo da camada de Aplicação do modelo OSI utilizado para transferência de dados na World Wide Web.

IPX – “Internet Packet Exchange” desenvolvido para suportar redes NetWare, suporta redes de tamanho pequeno e médio e também tem a capacidade básica de roteamento.

LAN – “Local Area Network” diz respeito à abrangência da rede de computadores, dita rede local.

MAN – “Metropolitan Área Network” rede de caráter metropolitano que liga computadores e usuários em uma área geográfica maior que a abrangida pela LAN mas menor que a abrangida pela WAN.

MIB – “Management Information Base” (Base para Administração de Informações). Ela contém um conjunto padrão de dados estatísticos e de controle, descrevendo o status do agente.

OSI – “Opens System Interconnect” padrão de conectividade para sistemas abertos, definido pela International Organization for Standardization. Composto por sete camadas diferentes para níveis diferentes de abstração no manuseio de dados.

PDU – “Protocol Data Units” comandos possibilitam a troca de informações entre gerente e o agente no sistema de gerenciamento, através deles o gerente tem a possibilidade de ler valores em variáveis da MIB e também armazenar.

RMON – “Remote Monitoring” é uma MIB padrão do protocolo SNMP, camada 4 do modelo OSI, que possui maior autonomia de funcionalidades em relação ao gerente, gerando em conseqüência menor tráfego na rede, indicado por isto para monitoramentos remotos.

SMTP – “Simple Mail Transfer Protocol” fornece um serviço store-and-forward para mensagens que carregam correspondências contendo textos;

SNMP – “Simple Network Management Protocol” protocolo de controle e monitoração de redes, camada de aplicação do protocolo TCP/IP, ou camada 4 do modelo OSI, amplamente utilizado. Serve para administrar configurações, estatísticas, performance e segurança em equipamentos de uma rede.

TCP/IP – “Transfer Control Protocol/Internet Protocol” protocolo de comunicação básico da internet, composto de quatro camadas: apliação, transporte, rede e físico. Utilizado na implementação de redes privativas como intranets e extranets.

TFTP – “Trivial File Transfer Protocol” versão simplificada do FTP que permite arquivos sejam transferidos de um computador a outro dentro de uma rede.

UDP – “User Datagram Protocol” protocolo da camada de transporte do modelo TCP/IP não orientado a conexão.

WAN – “Wide Area Network” quando uma rede cresce geograficamente para outros locais, sejam estados ou países, mantendo a conexão de seus usuários, dita rede de longa distância.

Capítulo 1 – Introdução

Este capítulo visa apresentar as motivações no desenvolvimento deste trabalho, mostrando qual a importância e contribuição, bem como contextualiza a sua área de abrangência.

1.1. Cenário

Por mais que haja projetos bem definidos de redes, mesmo sendo o tráfego corretamente estimado e os equipamentos dimensionados de acordo, com o passar do tempo, problemas como:

- Aumento de usuários a números não previstos;
- Aplicações demandarão mais recursos de rede do que se esperava;
- Desgaste dos elementos da camada física de rede.

Acarretarão congestionamentos, perda de pacotes de informação, queda do desempenho e disponibilidade e principalmente insatisfação dos usuários.

Os problemas ocorridos em uma rede de computadores são de vários tipos:

- Repetitivos;
- Inéditos;
- Repentinos.

A cada vez que acontecem implicam em prováveis paradas de serviço, afetando os usuários em seus trabalhos, gerando caos para os administradores que deixam seus afazeres para resolver problemas urgentes.

As redes heterogêneas estão aumentando sua presença, o que resulta em uma escala variada de:

- Serviços;
- Aplicações;
- Elementos de rede.

Cada qual com diferentes exigências de recursos e exigindo também conhecimento de suas peculiaridades pelo administrador.

Os administradores geralmente criam um centro de gerenciamento, onde ficam absorvidos em total dedicação, avaliando gráficos, comportamentos de variáveis através de ferramentas de monitoração.
Dependendo da ferramenta o número de variáveis apresentadas é grande, o que pode exigir experiência do administrador para conclusões de correções a serem feitas em parâmetros da rede.
Muitas vezes necessita ler relatórios de log's, nem sempre muito claros e amigáveis. Enfim deve estar atento a todos os sinais indicativos de problemas atuais ou futuros e sempre pronto para atuar rapidamente nas situações de alarmes.

Nas redes locais (LAN's) normalmente temos um administrador responsável, porém em redes metropolitanas (MAN'S) ou de longa distâncias (WAN's), teremos um número maior de administradores, o que pode gerar muitos conflitos de entendimentos e atuações.
Em muitas situações os problemas podem não se limitar apenas à rede local, e o consenso dos administradores será necessário para o rápido restabelecimento das condições normais, o que pode não acontecer de maneira tão fácil e rápida devido a provável discrepância de conhecimentos e experiência dos mesmos.

Há várias ferramentas no mercado que auxiliam o administrador em suas tarefas de gerência, porém poucas que utilizam a própria "inteligência" deste para resolver ou até prever os problemas.
Ferramentas que propiciassem uma maneira de armazenar o conhecimento e estendê-lo aos demais administradores não tão experientes, enfim que todos os envolvidos com a gerência da rede possam contribuir progressivamente no enriquecimento desta

ferramenta, a qual o quanto mais for utilizada mais "inteligente" ficará num processo tal qual um aprendizado.

O ponto ideal desta ferramenta seria quando ela contivesse uma Inteligência Artificial plena, ou seja, ficasse tão próxima do raciocínio humano no auxílio à gerência de redes, que não se pudesse, a tal ponto, definir se a solução apresentada para o problema foi apresentada pelo administrador ou pela ferramenta .

Uma das muitas definições de IA, que se aproxima muito do que foi desenvolvido neste trabalho diz: "*IA é uma parte da ciência da computação preocupada com sistemas computacionais inteligentes, isto é, sistemas que exibam características que associamos com o comportamento inteligente do ser humano (entender uma linguagem, ver, ouvir, aprender, raciocinar, resolver problemas etc)*".

Neste trabalho foi desenvolvida uma ferramenta tipo "shell", ou seja, os seus usuários, administradores, é que através da sua utilização e experiência terão condições de adicionar conteúdo e ajustar às condições específicas da rede a qual gerenciam. Isto quer dizer que este trabalho não pretende desenvolver uma "caixa preta" que prometa elucidar todos os problemas que possam ocorrer em determinada rede, mas sim encorajar o administrador da rede a participar na ação de tornar a ferramenta útil e poderosa.

1.2. Contexto do Trabalho

Da área da informática, este trabalho aborda redes de computadores, mais especificamente o tópico sobre gerência de redes. Também faz parte do contexto deste trabalho a área de Inteligência Artificial,mais especificamente sistema especialista.
A meta desta pesquisa é demonstrar a integração destas duas áreas a fim de implementar uma aplicação que monitore uma rede de computadores utilizando técnicas de Inteligência Artificial. Podendo ser uma rede de pequeno porte centralizada, até uma rede de grande porte descentralizada, desde que seja uma rede que atenda ao padrão OSI e que utilize o protocolo SNMP.

1.3. Problema

É comum encontrar ferramentas de gerência de redes que apresentam muitas variáveis de controle e várias técnicas para se garantir o comportamento dentro das expectativas.

Isto gera a necessidade do administrador da rede estar atento na maior parte do tempo a fim de corrigir configurações adotadas, cada vez mais em menor espaço de tempo.

Também nem sempre é escolhida a melhor configuração para aquele estado da rede, ou ainda, é adotada uma configuração que vigora na grande maioria das variações de performance da rede, levando a prejuízos temporários, que podem ser curtos e pouco percebidos ou longos e danosos à grande maioria dos usuários.

O administrador deve possuir um conhecimento elevado do comportamento da rede a qual gerencia, pois ele irá decidir os valores desejados para as variáveis de controle. Uma vez definidos estes valores, isto não quer dizer que o administrador poderá se despreocupar para as condições da rede. O administrador deve continuar acompanhando os indicadores de tela, as tendências a fim de evitar problemas de desempenho.

As aplicações para gerenciamento de redes existentes hoje são completas, oferecendo uma gama de informações elevada, o que muitas vezes vai exigir uma análise acurada de toda esta gama de dados pelo administrador da rede, e além disto há a necessidade, uma vez configurados os parâmetros, de um acompanhamento também criterioso do desempenho da rede face à parametrização adotada.
Pode-se dizer que uma parametrização adotada em um determinado momento da rede pode não ser ideal em um outro momento, já que as diversidades de aplicações existentes em uma rede são muito variadas, gerando exigências distintas de recursos. Isto implica em uma dedicação quase que exclusiva do administrador da rede a fim de manter o desempenho da rede em níveis de qualidade.

Em suma, por mais que existam ferramentas de gerenciamento, estas não garantem um desprendimento do administrador, e sim requerem um acompanhamento nas configurações dos valores dos parâmetros dos recursos da rede e a dedicação quase que exclusiva na monitoração dos alertas emitidos.
Neste caso temos um profissional de capacidade, por certo caro, dedicando uma boa quantidade de seu tempo com a preocupação de manter a rede em condições satisfatórias.

1.4. Motivação

A disseminação do uso de computadores vem crescendo muito nos últimos anos, praticamente qualquer atividade que se imagina está ou estará em breve suportada por computadores, não obstante estes não são simplesmente unidades de processamento estanques. Sim parte de um sistema maior chamado rede de computadores, onde há um compartilhamento de recursos tais como:

- Impressoras;
- Servidores;
- Softwares;
- E principalmente informação e comunicação.

Atualmente está havendo uma convergência das redes de telefonia pública e as redes de dados privadas, já é possível a conversação através dos computadores.

A dependência destas atividades, do perfeito funcionamento da rede de computadores também aumentou consideravelmente. Imagine um usuário que não consiga ler ou enviar um e-mail, em muitos casos praticamente o seu trabalho é paralisado. Ou empresas que utilizam aplicativos para automatizar seus processos internos, uma rede fora de operação pode trazer percalços e prejuízos enormes.

Neste cenário exigente, surgiu a figura o administrador de redes, com a incumbência de configurar e monitorar os parâmetros das redes, a fim de que qualquer problema que

venha a ocorrer seja resolvido imediatamente, mas preferencialmente que os problemas sejam previstos e evitados.

Em auxílio a esta árdua tarefa os fabricantes de equipamentos desenvolveram uma miríade de ferramentas que facilitam as atividades do administrador. Há até dificuldade em decidir qual dentre tantas escolher. Uma vez escolhida a ferramenta, persiste ainda uma dificuldade que é configura-la, quando possível, para que esta se adeqüe ao que realmente precisa ser monitorado.

1.4.1. Justificativa

A ferramentas existentes no mercado são pacotes fechados, na maior parte das vezes tem muito mais do que se precisa, ou em outros casos, não contempla um parâmetro que seria de vital importância para determinado tipo de rede. Não interagem com o administrador, não utilizam sua “expertise”, não são pró-ativas.

Este cenário instigou o desenvolvimento de uma ferramenta, que venha a suprir estas deficiências e principalmente que seja fácil de utilizar e interagir, com a qual o administrador possa utilizar seu conhecimento acumulado como também as particularidades da rede a qual administra para enriquecer a ferramenta, adaptando-a às suas reais necessidades.

1.5. Objetivos

Convém que seja feita uma divisão do tópico objetivos. Há o âmbito geral, mais amplo, que culmina com o desenvolvimento de uma aplicação, mas não podemos deixar de citar os objetivos que foram alcançados durante cada etapa deste trabalho, também com alto grau de importância.

1.5.1. Objetivo Geral.

O objetivo deste trabalho é desenvolver uma ferramenta para a monitoração de redes, utilizando-se de técnicas de gerência de redes e de Inteligência Artificial, mais especificamente os Sistemas Especialistas. Este sistema é uma "Shell", a qual contempla uma base de conhecimento inicial com alguns parâmetros relevantes na maioria das redes.
Seu diferencial em relação às demais ferramentas, é que possibilita ao usuário selecionar, dentro de um conjunto pré-definido, outros parâmetros que considere serem importantes para a rede a qual monitora. Com isto, um conhecimento de consenso nesta base, vai ser modificado e/ou inserido e acompanhado, conforme o especialista desejar.

A proposta é que seja uma ferramenta de apoio a decisão, com um conhecimento inicial, mas que o especialista poderá interagir de tal forma que ela possa conter todo o conhecimento do próprio administrador, o que permitirá a este, em muitos momentos, a possibilidade de delegar a administração.

O objetivo deste trabalho também foi mostrar a interação de duas áreas da ciência, a gerência de redes e a Inteligência Artificial, provando que é perfeitamente possível desenvolver sistemas funcionais, que atendam às expectativas do usuário e mais, que o torne um partícipe da escalabilidade da ferramenta, propiciando a este a interação com o sistema, tornando-o adaptativo às suas necessidades.

1.5.2. Objetivos Específicos

- Desenvolver ferramenta com forte interação e adaptação às necessidades dos usuários;
- Utilizar técnicas de Inteligência Artificial em apoio à administração de redes;
- Implementação de um programa na linguagem C para ler variáveis da MIB;
- Implementar uma "Shell" de sistema especialista;
- Integrar o programa em C à "Shell" gerando um protótipo;
- Testar o protótipo através de simulações.

1.6. Organização do Trabalho.

O trabalho foi organizado de maneira a inicialmente abordar os tópicos referentes à gerência de redes, a fim de fornecer o embasamento necessário para o entendimento da da interface SNMP e dos parâmetros que serão monitorados, isto abrange os capítulos 2, 3, 4 e 5. No capítulo 6 a área de Inteligência Artificial é abordada em seu tópico sistemas especialistas, que é suficiente para o entendimento do funcionamento da "Shell" que será implementada na linguagem Prolog. No capítulo 7 parte-se para a implementação da proposta, onde foi desenvolvida uma interface SNMP na linguagem C, para gerar um arquivo atualizado de variáveis e valores, que será lido pela outra implementação desenvolvida em Prolog, que faz o papel do monitor da rede. No capítulo 8 é exibida a conclusão sobre a expectativa inicial e os resultados obtidos, bem como sugestões de trabalhos futuros.

Pelo exposto neste capítulo conclui-se que este trabalho não se trata apenas de mais uma ferramenta para monitorar a rede, mas sim de uma ferramenta abrangente com uma proposta forte de interação e adaptação às necessidades dos usuários.

Capítulo 2 – Gerenciamento de Redes

Neste capítulo é apresentada uma classificação da gerência de redes a fim de identificar exatamente qual a intenção de gerência deste trabalho. Diante desta classificação percebe-se que o assunto é bastante amplo. Também é mostrado o conteúdo de um modelo de sistema de gerência, a fim de identificar as partes importantes para a construção de uma ferramenta de gerência.

2.1. Contexto

O contínuo crescimento em número e diversidade dos componentes das redes de computadores tem tornado a atividade de gerenciamento da rede cada vez mais complexa. Duas causas principais têm tornado árduo o trabalho de isolamento e teste de problemas [Tarouco93]:

- Diversidade dos níveis do pessoal envolvido: técnicos, gerentes e engenheiros;
- Diversidade nas formas de controle e monitoração: produtos cada vez mais complexos, cada fornecedor oferecendo ferramentas próprias de controle e monitoração.

As atividades básicas do gerenciamento de redes consistem na detecção e correção de falhas em um tempo mínimo e no estabelecimento de procedimentos para a previsão de problemas futuros. Por exemplo, é possível tomar medidas que evitem o colapso da rede, como a reconfiguração das rotas ou a troca do roteador por um modelo mais adequado, através da monitoração de linhas cujo tráfego esteja aumentando ou roteadores que estão se sobrecarregando.

A eficiência na realização destas tarefas está diretamente ligada a presença de ferramentas que as automatizem e de pessoal qualificado. Atualmente existem no mercado diversos tipos de ferramentas que auxiliam o administrador nas atividades de gerenciamento. Estas ferramentas podem ser divididas em quatro categorias [Oda94a]:

- Ferramentas de nível físico, que detectam problemas em termos de cabos e conexões de *hardware*;
- Monitores de rede, que se conectam às redes monitorando o tráfego;
- Analisadores de rede, que auxiliam no rastreamento e correção de problemas encontrados nas redes;
- Sistemas de gerenciamento de redes, os quais permitem a monitorização e controle de uma rede inteira a partir de um ponto central.

Dentre a gama de soluções possíveis para o gerenciamento de redes, uma das mais usuais consiste em utilizar um computador que interage com os diversos componentes da rede para deles extrair as informações necessárias ao seu gerenciamento.

Evidentemente é preciso montar um banco de dados neste computador que será gerente da rede, contendo informações necessárias para apoiar o diagnóstico e a busca de soluções para problemas da rede. Isto envolve esforço para identificar, rastrear e resolver situações de falhas. Como o tempo de espera do usuário pelo restabelecimento do serviço deve ser o menor possível, tudo isto deve ser feito eficientemente.

A seguir são mostradas as várias formas de gerenciamento, para situar onde a proposta deste trabalho se situa.

2.2. Gerenciamento de Falhas

Uma falha é uma condição anormal cuja recuperação exige ação de gerenciamento. Normalmente é causada por operações incorretas ou um número excessivo de erros. Por exemplo, se uma linha de comunicação é cortada fisicamente, nenhum sinal pode passar através dela. Um grampeamento no cabo pode causar distorções que induzem a uma alta taxa de erros. Certos erros como, por exemplo, um bit errado em uma linha de comunicação, pode ocorrer ocasionalmente e normalmente não é considerada falha.

Para controlar o sistema como um todo, cada componente essencial deve ser monitorado individualmente para garantir o seu perfeito funcionamento. Quando ocorre

uma falha, é importante que seja possível, rapidamente:

- Determinar o componente exato onde a falha ocorreu;
- Isolar o resto da rede da falha, de tal forma que ela continue a funcionar sem interferências;
- Reconfigurar ou modificar a rede para minimizar o impacto da operação sem o componente que falhou;
- Reparar ou trocar o componente com problemas para restaurar a rede ao seu estado anterior.

O impacto e a duração do estado de falha pode ser minimizado pelo uso de componentes redundantes e rotas de comunicação alternativas, para dar à rede um grau de "tolerância à falhas" [Specialski 2001].

O presente trabalho irá atuar na identificação do componente, elemento de rede, que está gerando a falha para o sistema, logo esta classificação de gerência importa para a ferramenta desenvolvida neste trabalho. Porém vamos seguir verificando as demais possibilidades.

2.3. Gerenciamento de Contabilização

Mesmo que nenhuma cobrança interna seja feita pela utilização dos recursos da rede, o administrador da rede deve estar habilitado para controlar o uso dos recursos por usuário ou grupo de usuários, com o objetivo de:

- Evitar que um usuário ou grupo de usuários abuse de seus privilégios de acesso e monopolize a rede, em detrimento de outros usuários;
- Evitar que usuários façam uso ineficiente da rede, assistindo-os na troca de procedimentos e garantindo a desempenho da rede;
- Conhecer as atividades dos usuários com detalhes suficientes para planejar o crescimento da rede.

O gerente da rede deve ser capaz de especificar os tipos de informações de contabilização que devem ser registrados em cada nodo, o intervalo de entrega de relatórios para nodos de gerenciamento de mais alto nível e os algoritmos usados no cálculo da utilização [Specialski 2001].

A gerência de contabilização trata mais da configuração do perfil dos usuários do que propriamente com os elementos de rede, por esta razão não interfere na proposta deste trabalho.

2.4. Gerenciamento de Configuração

O gerenciamento de configuração está relacionado com a inicialização da rede e com uma eventual desabilitação de parte ou de toda a rede. Também está relacionado com as tarefas de manutenção, adição e atualização de relacionamentos entre os componentes e do status dos componentes durante a operação da rede.

Alguns recursos podem ser configurados para executar diferentes serviços como, por exemplo, um equipamento que pode atuar como roteador, como estação de trabalho ou ambos. Uma vez decidido como o equipamento deve ser usado, o gerente de configuração pode escolher o software apropriado e um conjunto de valores para os atributos daquele equipamento.

O gerente da rede deve ser capaz de, inicialmente, identificar os componentes da rede e definir a conectividade entre eles. Também deve ser capaz de modificar a configuração em resposta à avaliações de desempenho, recuperação de falhas, problemas de segurança, atualização da rede ou a fim de atender à necessidades dos usuários.

Relatórios de configuração podem ser gerados periodicamente ou em resposta à requisições de usuários [Specialski 2001].

O gerenciamento de configuração por tratar diretamente dos elementos de rede, tem relação com a gerência de falhas, um roteador mau configurado pode gerar muita fragmentação de pacotes de dados, que deverá ser detectado pela gerência de falhas. O tópico seguinte também é importante para este trabalho.

2.5. Gerenciamento de Desempenho

O gerenciamento do desempenho de uma rede consiste na monitoração das atividades da rede e no controle dos recursos através de ajustes e trocas. Algumas das

questões relativas ao gerenciamento do desempenho, são:

- Qual é o nível de capacidade de utilização?
- O tráfego é excessivo?
- O “throughput” foi reduzido para níveis aceitáveis?
- Existem gargalos?
- O tempo de resposta está aumentando?

Para tratar estas questões, o gerente deve focalizar um conjunto inicial de recursos a serem monitorados, a fim de estabelecer níveis de desempenho. Isto inclui associar métricas e valores apropriados aos recursos de rede que possam fornecer indicadores de diferentes níveis de desempenho. Muitos recursos devem ser monitorados para se obter informações sobre o nível de operação da rede. Colecionando e analisando estas informações, o gerente da rede pode ficar mais e mais capacitado no reconhecimento de situações indicativas de degradação de desempenho.

Estatísticas de desempenho podem ajudar no planejamento, administração e manutenção de grandes redes. Estas informações podem ser utilizadas para reconhecer situações de gargalo antes que elas causem problemas para o usuário final. Ações corretivas podem ser executadas, tais como, trocar tabelas de roteamento para balancear ou redistribuir a carga de tráfego durante horários de pico, ou ainda, a longo prazo, indicar a necessidade de expansão de linhas para uma determinada área [Specialski 2001].

O desempenho é importante quando se fala em termos de gerência, pois avalia os parâmetros dinâmicos da rede, porém será visto no capítulo sobre o protocolo SNMP que há limitações de versões do protocolo para atender a estas necessidades de informações. Por fim vejamos a última classificação de gerenciamento.

2.6. Gerenciamento de Segurança

O gerenciamento da segurança provê facilidades para proteger recursos da rede e informações dos usuários. Estas facilidades devem estar disponíveis apenas para usuários autorizados. A política de segurança deve ser robusta e efetiva e que o sistema

de gerenciamento da segurança seja, ele próprio, seguro.

O gerenciamento de segurança trata de questões como:

- Geração, distribuição e armazenamento de chaves de criptografia;
- Manutenção e distribuição de senhas e informações de controle de acesso;
- Monitoração e controle de acesso à rede ou parte da rede e às informações obtidas dos nodos da rede;
- Coleta, armazenamento e exame de registros de auditoria e logs de segurança, bem como ativação e desativação destas atividades [Specialski 2001].

Aqui novamente a preocupação é com o perfil dos usuários, como na gerência de contabilização, certos privilégios são dados a alguns usuários e restrições são dadas a outros, também pouco interfere com a proposta deste trabalho. A seguir é dada uma visão mais geral dos elementos envolvidos na gerência.

2.7. Sistemas de Gerenciamento de Redes

Os sistemas de gerenciamento de redes apresentam a vantagem de ter um conjunto de ferramentas para análise e depuração da rede. Estes sistemas podem apresentar também uma série de mecanismos que facilitam a identificação, notificação e registro de problemas, como por exemplo [Oda94a]:

- Alarmes que indicam, através de mensagens ou bips de alerta, anormalidades na rede;
- Geração automática de relatórios contendo as informações coletadas;
- Facilidades para integrar novas funções ao próprio sistema de gerenciamento;
- Geração de gráficos estatísticos em tempo real;
- Apresentação gráfica da topologia das redes.

Em redes IP, o sistema de gerenciamento segue o modelo gerente-agente, onde o GERENTE é o próprio sistema de gerenciamento e o AGENTE é um *software* que deve ser instalado nos equipamentos gerenciados. A tarefa do agente é responder as

requisições feitas pelo gerente em relação ao equipamento no qual o agente está instalado. Esta interação é viabilizada pelo protocolo de gerenciamento "*Simple Network Management Protocol"* (SNMP), o qual é como uma linguagem comum utilizada exclusivamente para a troca de informações de gerenciamento. Dessa forma, o gerente consegue conversar com qualquer máquina que fale SNMP, independente do tipo de "*hardware"* e sistema operacional. O conjunto de informações ao qual o gerente pode fazer requisições ou alterações é denominado de "*Management Information Base"* (MIB).

É importante ressaltar que o protocolo SNMP é adotado como padrão na Internet pela sua simplicidade. A funcionalidade reside basicamente na leitura e alteração de valores de variáveis. O modelo OSI utiliza um protocolo complexo, o CMIP (Common Management Information Protocol), que apresenta funcionalidades para a execução de operações sobre vários objetos a partir da definição de um escopo e de um filtro para a seleção destes objetos.

O protocolo de gerenciamento SNMP constitui atualmente um padrão operacional "de facto", e grande parte do seu sucesso se deve a sua simplicidade, sendo um protocolo "*send*/*receive"* com apenas quatro operações. Outro aspecto importante é a sua capacidade de gerenciar redes heterogêneas constituídas de diferentes tecnologias, protocolos e sistemas operacionais. Dessa forma, o SNMP pode gerenciar, por exemplo, redes Ethernet, Token Ring e FDDI, conectando IBM PCs, Apple Machintosh, estações de trabalho SUN e outros tipos de computadores [Oda94b].

As aplicações de gerenciamento utilizam o SNMP para [Oda94b] :

- Fazer "*polling"* nos dispositivos de rede e coletar dados estatísticos para análise em tempo real;
- Receber um conjunto limitado de notificações de eventos significativos ou mensagens "*trap"*;
- Reconfigurar dispositivos de rede.

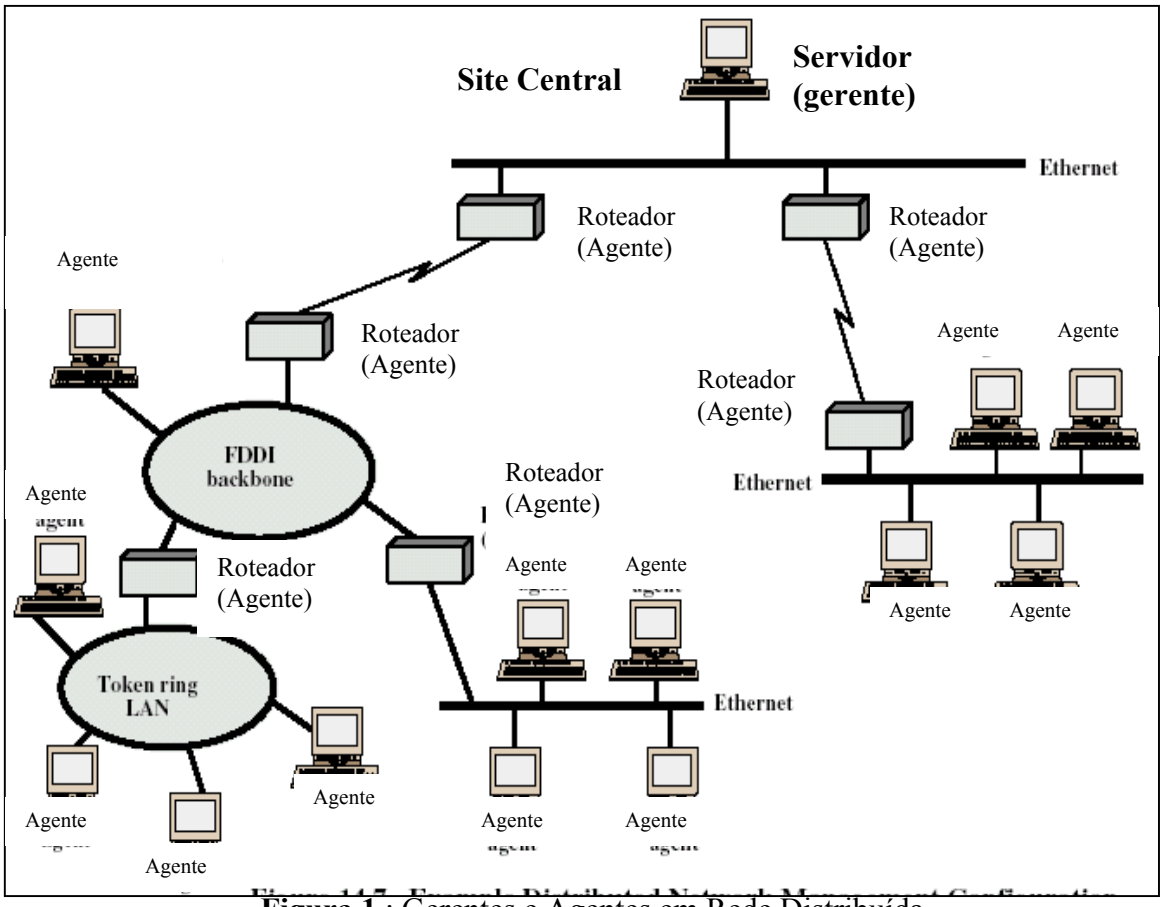


Figura 1 : Gerentes e Agentes em Rede Distribuída.

A figura acima exemplifica um ambiente distribuído de gerência, integrando várias topologias de rede: FDDI, Tokeng ring e Ethernet. Percebe-se também que cada elemento de rede ou é um agente, em maior número, ou é um gerente e até um caso onde o elemento de rede pode assumir os dois papeis. No capítulo sobre o protocolo SNMP este assunto vai ser detalhado.

Das classificações apresentadas neste capítulo, a que será utilizada neste trabalho é a gerência de falhas. A preocupação da ferramenta que foi desenvolvida é em possibilitar que possíveis falhas com elementos de rede sejam evitadas, melhorando a disponibilidade e performance da rede.

No capítulo 3 os conceitos de gerente, agente, elementos de rede e o protocolo SNMP serão explorados, ficando mais clara a interação de todos estes elementos.

Capítulo 3 – Protocolo SNMP

O SNMP é um protocolo da camada de aplicação do modelo TCP/IP, designado para facilitar a troca de informações de gerenciamento entre dispositivos de rede. Se considerássemos o modelo OSI o TCP estaria inserido na camada 4 e o IP na camada 3. Usando os dados transportados pelo SNMP, os administradores de rede podem gerenciar mais facilmente a performance da rede, encontrar e solucionar problemas e planejar com mais precisão uma possível expansão da rede.

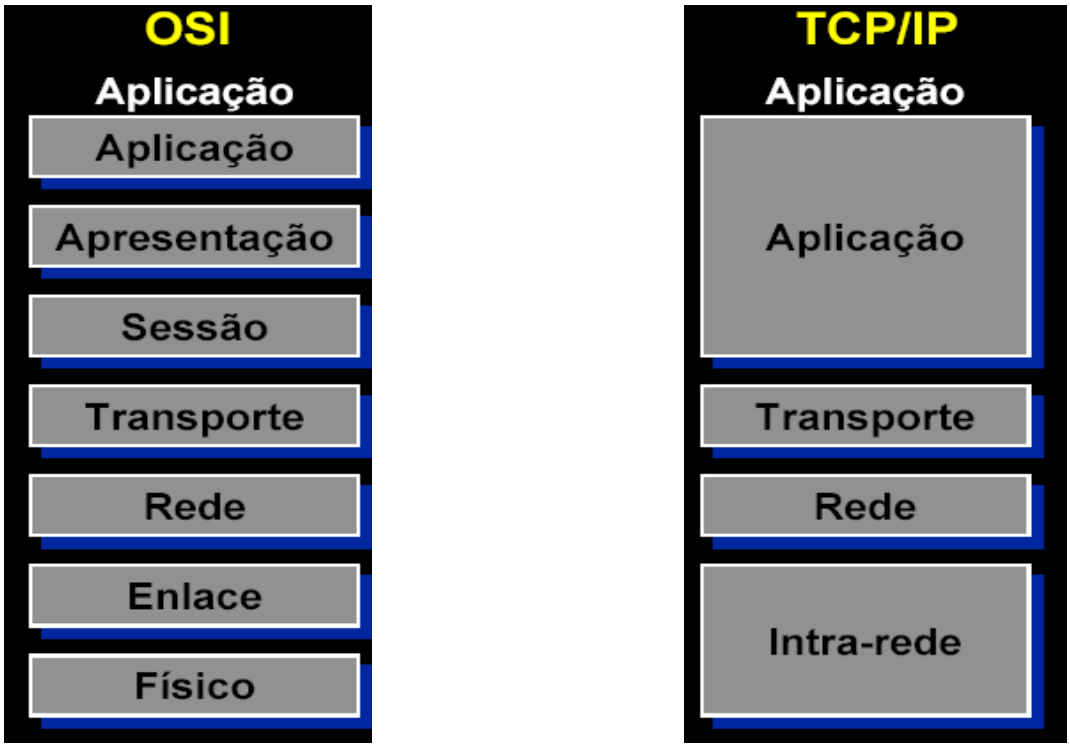


Figura 2: Estrutura do Protocolo TCP/IP localizando o SNMP.

Atualmente, o SNMP é o mais popular protocolo para gerenciamento de diversas redes comerciais como as usadas em universidades, centros de pesquisas e provedores de acesso e de informações. Esta popularização se deve ao fato de que o SNMP é um

protocolo relativamente simples, porém suficientemente poderoso para resolver os difíceis problemas apresentados quando se tenta gerenciar redes heterogêneas.

Portanto neste capítulo são exploradas as características importantes deste protocolo, que serão utilizadas no desenvolvimento da interface entre o agente e o gerente.

3.1. Modelo de Gerenciamento SNMP

O gerenciamento SNMP é baseado na interação de diversas entidades, como se segue [Case90, Oda94b]:

- Elementos de rede: também chamados *dispositivos gerenciados*, os elementos de rede são dispositivos de *hardware* como os computadores, roteadores, e servidores de terminais que estão conectados a rede;
- Agentes: são módulos de *software* que residem nos elementos de rede. Eles coletam e armazenam informações de gerenciamento como o número de pacotes de erros recebidos pelo elemento de rede, respondem às solicitações dos gerentes;
- Objeto gerenciado: um objeto gerenciado é qualquer elemento que possa ser gerenciado. Por exemplo, uma lista dos circuitos TCP atualmente ativos em um "*host* "particular é um objeto gerenciado;
- MIB: uma MIB é uma coleção de objetos gerenciados residentes em um armazenamento virtual de informações. Coleções de objetos gerenciados relacionados são definidas em módulos específicos da MIB;
- Notação sintática: é a linguagem usada para descrever os objetos gerenciados da MIB em um formato independente da plataforma. Um uso consistente da notação sintática permite que diferentes tipos de computadores compartilhem informações. Sistemas de gerenciamento Internet usam um subconjunto da "*Open System Interconnection"* (OSI), "*Abstract Syntax Notation 1"* (ASN.1) da "*International Organization for Standardization's"* (ISO) para definir tanto

os pacotes que são trocados pelo protocolo de gerenciamento quanto os objetos que ele deve gerenciar;

- *"Structure of Management Information"* (SMI): o SMI define as regras para descrever as informações de gerenciamento. O SMI é definido usando ASN.1;
- *"Network Management Stations"* (NMS): também chamados consoles, estes dispositivos executam aplicações de gerenciamento para monitorar e controlar elementos de rede. Fisicamente, os NMS são usualmente "*workstations"* com CPU velozes, monitores coloridos de alta definição, memória substancial e um grande espaço em disco;
- Protocolo de gerenciamento - um protocolo de gerenciamento é usado para transportar informações de gerenciamento entre agentes e NMS. O SNMP é o protocolo de gerenciamento padrão da comunidade Internet.

Com base nesta arquitetura, o SNMP foi construído para minimizar a quantidade e a complexidade das funções necessárias para gerenciar um agente [Case90].

A figura 3 demonstra a dinâmica do protocolo SNMP sobre as camadas do protocolo TCP/IP, do lado esquerdo está uma estação gerenciadora, onde é executada uma aplicação de gerenciamento, que através das primitivas de comunicação do protocolo (GetRequest, SetRequest etc.) descendo as camadas do TCP/IP, interage com o agente, bloco da direita, subindo as camadas do TCP e acessando os objetos gerenciados, que são na verdade parâmetros de rede instanciados na base (MIB) do agente e atualizados por este.

Figura 3: Dinâmica do Protocolo SNMP sobre o TCP/IP.

Os processos que implementam as funções de gerenciamento Internet atuam ou como agentes ou como gerentes. Os agentes coletam junto aos dispositivos gerenciados as informações relevantes ao gerenciamento da rede. O gerente, por sua vez, processa essas informações com o objetivo de detectar falhas no funcionamento dos elementos da rede, para que "*possam ser tomadas providências no sentido de contornar os problemas que ocorrem como conseqüência das falhas*" [Soares97].

Um objeto gerenciado representa um recurso e pode ser visto como uma coleção de variáveis cujo valor pode ser lido ou alterado. Para tanto o gerente envia comandos aos agentes. Para monitorar os dispositivos gerenciados, o gerente solicita ao agente uma leitura no valor das variáveis mantidas por estes dispositivos, através do comando "*Get"*, e o agente responde através do comando "*Response"*.

Para controlar os dispositivos gerenciados, o gerente modifica o valor das variáveis armazenadas nos dispositivos gerenciados, através do comando "*Put"*. Isto pode ser

usado para disparar indiretamente a execução de operações nos recursos associados aos objetos gerenciados. Por exemplo, um "reboot" do elemento de rede pode ser facilmente implementado, basta que o gerente modifique o parâmetro que indica o tempo até uma reinicialização do sistema.

- gerente pode ainda determinar que variável um dispositivo gerenciado suporta e colher informações de forma seqüencial, das tabelas de variáveis (como as tabelas de roteamento IP) nos dispositivos gerenciados. Para isto, ele utiliza as operações transversais ("*transversal operations"*).

Em alguns casos é necessário que a troca de informações seja em sentido inverso, isto é, o agente tem de passar informações para o gerente. O SNMP define a operação "*Trap"* para que um agente informe ao gerente a ocorrência de um evento específico [Cisco96].

3.2. Definição dos Relacionamentos Administrativos

A arquitetura SNMP admite uma variedade de relacionamentos administrativos entre entidades que participam do protocolo. As entidades residentes nas estações gerenciadas e os elementos de rede que se comunicam com um outro elemento usando SNMP são chamados de entidades de aplicação SNMP. O processo que implementa o SNMP, e portanto suporta as entidades de aplicação SNMP, é chamado protocolo de entidades.

A junção de um agente SNMP com algum conjunto arbitrário de entidades de aplicação SNMP é chamada de comunidade SNMP. Cada comunidade é nomeada através de uma cadeia de octetos.

Uma mensagem SNMP, originada por uma entidade de aplicação SNMP que de fato pertence a comunidade SNMP referenciada pela mensagem, é chamada mensagem SNMP autêntica. O conjunto de regras existentes para que uma mensagem seja identificada como uma mensagem SNMP autêntica para uma comunidade SNMP qualquer é chamado de esquema de autenticação. A implementação de uma função que

identifica mensagens autênticas de acordo com um ou mais esquemas de autenticação é chamado serviço de autenticação.

Evidentemente, um efetivo gerenciamento das relações administrativas entre entidades de aplicação SNMP requer que os serviços de autenticação (pelo uso de criptografia ou outra técnica) sejam capazes de identificar mensagens autênticas com um alto grau de confiabilidade.

Para qualquer elemento da rede, um subconjunto de objetos na MIB que pertence a este objeto é chamado de visão da MIB SNMP. Um elemento do conjunto (READ-ONLY, READ-WRITE) é chamado de modo de acesso SNMP.

A junção do modo de acesso SNMP com a visão da MIB é chamada de perfil da comunidade SNMP. O perfil da comunidade SNMP representa um privilégio de acesso específico para variáveis em uma MIB específica. A união da comunidade SNMP com o perfil da comunidade é chamada de política de acesso SNMP. Uma política de acesso representa um perfil de comunidade específico proporcionado por um agente SNMP de uma comunidade para outros membros desta comunidade. Todos os relacionamentos administrativos entre entidades de aplicação SNMP são definidos em termos das políticas de acesso.

Para toda política de acesso SNMP, se o elemento de rede em que o agente SNMP especificado pela comunidade SNMP reside não contém a visão MIB que o perfil especifica, então esta política é chamada política de acesso "*proxy"* SNMP. O agente associado com a política de acesso "*proxy"* é chamado de agente "*proxy"*.

A política "*proxy* " é usualmente definida de duas maneiras :

- Ela permite a monitoração e o controle dos elementos de rede que não são endereçáveis usando o protocolo de gerenciamento e o protocolo de transporte. Um agente "*proxy"* provê uma função de conversão de protocolo permitindo a uma estação de gerenciamento aplicar um gerenciamento consistente em todos os elementos da rede, incluindo dispositivos como modems, multiplexadores, e outros dispositivos que suportam diferentes estruturas de gerenciamento;

- Ela potencialmente protege os elementos da rede de elaboradas políticas de controle de acesso. Por exemplo, um agente "*proxy*" pode implementar sofisticados controles de acesso, fazendo com que diversos subconjuntos de variáveis dentro de uma MIB se tornem acessíveis para diferentes estações de gerenciamento da rede, sem aumentar a complexidade do elemento de rede.

Este sub-capítulo apresentou aspectos de segurança do protocolo, o conceito de comunidade é presente no SNMP, porém esta autenticação é fraca, não se pode garantir segurança através de um octeto no cabeçalho de um pacote. A seguir será apresentado um tópico sobre as mensagens SNMP, que foi utilizado para o desenvolvimento da interface SNMP deste trabalho.

3.3. Operações SNMP

O SNMP por si só é um protocolo de requisição/resposta simples. Os NMS podem enviar múltiplas requisições sem receber uma resposta. Quatro operações são definidas no SNMP [Cisco96]:

- *Get*: permite que o NMS recupere uma instância de objeto do agente;
- *GetNext*: permite que o NMS recupere a próxima instância de objetos de uma tabela ou lista em um agente. Se o NMS quiser recuperar todos os elementos de uma tabela de um agente, ele inicia com uma operação *Get* seguida de uma série de operações *GetNext*;
- *Set*: permite que o NMS modifique valores de uma instância de objetos em um agente;
- *Trap*: usado pelo agente para informar assincronicamente o NMS sobre algum evento.

Os pacotes de mensagem do SNMP são divididos em duas partes. A primeira parte contém a versão e o nome comunitário. A segunda parte contém o protocolo de unidade de dados (PDU) do SNMP especificando a operação que será realizada (*"Get*", "*Set*" e

outros) e a instância de objetos envolvida na operação. A figura 4 ilustra o formato das mensagens no SNMP.

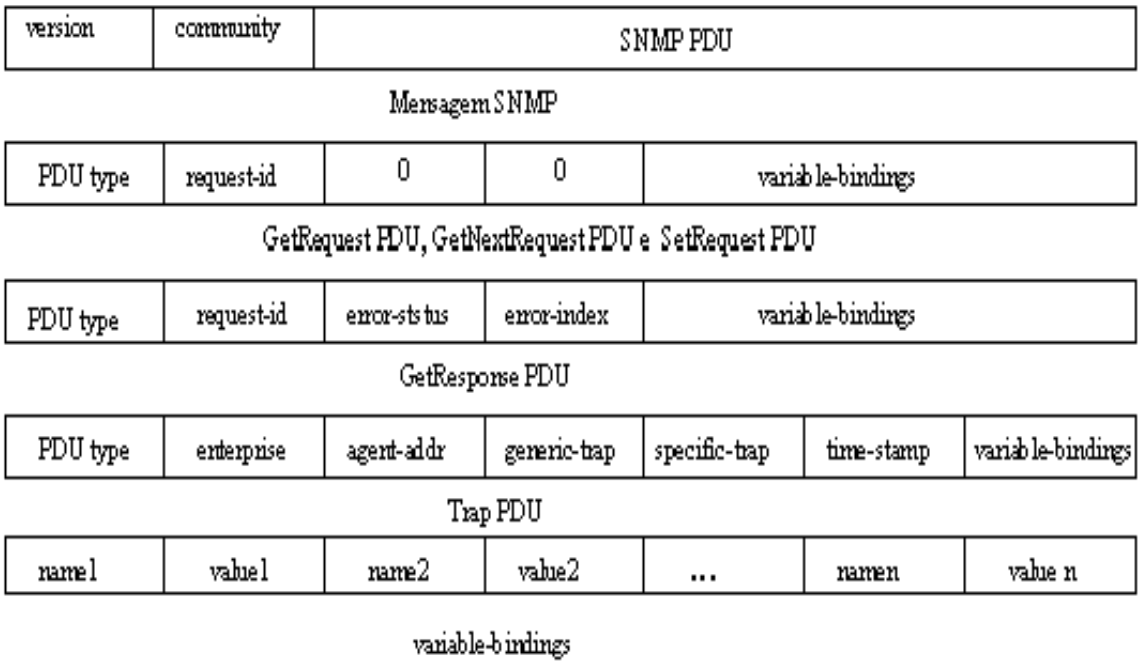


Figura 4 - Formato das Mensagens SNMP

O campo versão é usado para garantir que todos os elementos de uma rede estão rodando "*softwares"* baseados na mesma versão do SNMP. O nome comunitário associa um ambiente de acesso para um conjunto de NMS usando o nome comunitário. Um NMS dentro de uma comunidade pode ser dito existente dentro de um mesmo domínio administrativo. Como os dispositivos que não conhecem seu próprio nome comunitário são excluídos das operações do SNMP, então o administrador de redes usa o nome comunitário como uma forma fraca de autenticação.

O PDU do SNMP tem os seguintes campos [Cisco96] :

- Tipo PDU: especifica o tipo que o PDU começará transmitindo;
- Identificação de requisição: associa as requisições com as respostas;
- Variáveis ligadas: incluir o dado em um PDU SNMP. Variáveis ligadas associam instâncias particulares de objetos com seus valores correntes.

Este capítulo mostrou o ponto principal do protocolo SNMP, que é facilidade de geração de mensagens, chamadas operações SNMP, entre entidades de rede, as três mais importantes são: Get, Set, Trap, através da utilização destas três requisições pode-se construir uma boa ferramenta de gerência de redes.

O próximo capítulo explora a maneira de contornar uma das poucas deficiências do SNMP, que é o volume de dados que são trocados entre agente e gerente, que pode causar problemas para redes que trabalham no limite do tráfego. A solução foi a criação de uma monitoração remota.

Capítulo 4 - Monitoramento Remoto (RMON)

Este capítulo apresenta uma variação do protocolo SNMP, desenvolvida para contornar a necessidade que tem o gerente de ficar constantemente comunicando com o gerente. Também apresenta uma evolução do próprio RMON, chamado RMON2, que incrementa a segurança e o número de objetos gerenciados.

O protocolo SNMP não é adequado para ambientes de redes corporativas constituídas de diversas redes locais conectadas através de outra de longa distância. Esses enlaces de rede de longa distância, por operarem a taxas de transmissão inferiores às LAN's que a interconectam, passam a ter grande parte da sua banda de transmissão ocupada para informações de gerenciamento. Uma solução encontrada para dirimir este problema foi o "*Remote MONitoring"* (RMON), ou monitoramento remoto.

O protocolo RMON [Waldbusser91, Carvalho97], oferece suporte à implementação de um sistema de gerenciamento distribuído. Nele, fica atribuída aos diferentes elementos, tais como estações de trabalho, "*hubs"*, "*switches"* ou roteadores, das redes locais remotas a função de monitor remoto. Cada elemento RMON tem, então, como tarefas, coletar, analisar, tratar e filtrar informações de gerenciamento da rede e apenas notificar à estação gerente os eventos significativos e situações de erro. No caso de existirem múltiplos gerentes, cada elemento RMON deve determinar quais informações de gerenciamento devem ser encaminhados para cada gerente.

Sendo assim, os objetivos do protocolo RMON são [Carvalho97] :

- Reduzir a quantidade de informações trocadas entre a rede local gerenciada e a estação gerente conectada a uma rede local remota;
- Possibilitar o gerenciamento contínuo de segmentos de redes locais, mesmo quando a comunicação entre o elemento RMON e a estação gerente estiver, temporariamente, interrompida;
- Permitir o gerenciamento pró-ativo da rede, diagnosticando e registrando eventos que possibilitem detectar o mau funcionamento e prever falhas que interrompam sua operação;

- Detectar, registrar e informar à estação gerente, condições de erro e eventos significativos da rede;
- Enviar informações de gerenciamento para múltiplas estações gerentes, permitindo, no caso de situações críticas de operação da rede gerenciada, que a causa da falha ou mau funcionamento da rede possa ser diagnosticada a partir de mais de uma estação gerente.

Embora o RMON pareça oferecer grandes vantagens em relação ao SNMP, ainda não é muito fácil encontrar aplicações deste protocolo, a grande maioria da aplicações ainda são do protocolo SNMP puro. Vejamos então quais são os objetivos do RMON.

4.1. Objetivos do Protocolo RMON

Os dispositivos de monitoramento remoto de rede (RMON) são instrumentos que existem com o propósito de gerenciar uma rede. Muitas vezes estes dispositivos são dedicados e devotam significativos recursos internos para o propósito de gerenciamento de rede.

Uma organização pode dispor de diversos destes dispositivos, um por segmento de rede, para gerenciar sua rede Internet.

A especificação RMON é uma definição de uma MIB. O objetivo, contudo, é definir padrões de monitoração e interfaces para a comunicação entre agentes/gerentes SNMP.

Os objetivos da implementação de um dispositivo RMON, são os que se seguem [Waldbusser91] :

- Operação *Off-Line*: esta é a condição em que a estação gerenciadora não está em contato constante com o dispositivo RMON. Presta-se para desenhar redes de baixo custo de comunicação (redes por acesso discado ou conexões com "*World Area Networks"* - WAN) ou para acidentes onde as falhas na rede afetam a comunicação entre a estação gerenciadora e os dispositivos RMON. Desta forma, o monitor é configurado para coletar estatísticas e fazer diagnósticos

continuamente, mesmo se a conexão com o gerente não for possível ou apresentar falhas. O monitor pode também notificar a estação de gerenciamento se eventos excepcionais ocorrerem;

- Monitoramento Preemptivo: se o monitor tiver recursos disponíveis, estes podem ser usados para executar diagnósticos continuamente e para analisar a performance da rede. Quando uma falha ocorrer, o monitor pode notificar a estação de gerenciamento e armazenar o histórico estatístico referente à falha. Posteriormente este histórico pode ser enviado a estação de gerenciamento com o objetivo de se fazer um estudo mais profundo e permitir a detecção e reparo da falha;

- Detecção de Problemas e Geração de Relatórios: o monitor pode ser configurado para reconhecer certas situações, mais notadamente condições de erro e checar continuamente por elas. Quando uma destas situações ocorrer, o monitor pode registrá-la e reportá-la a estação de gerenciamento;

- Análise de Dados - por ser um dispositivo dedicado exclusivamente ao gerenciamento de rede e por estar localizado diretamente no segmento monitorado da rede, os dispositivos de monitoramento remoto (RMON) podem fazer uma análise significativa dos dados que coletam. Por exemplo, estes dispositivos podem determinar qual *host* gera maior tráfego ou mais erros na rede;

- Múltiplos Gerentes - uma configuração de rede pode ter mais de uma estação gerente para dar mais confiabilidade, executar funções diferentes e prover capacidades de gerência para unidades diferentes dentro da organização.

Percebe-se pelo exposto acima que a versatilidade do protocolo RMON é superior ao do SNMP, porém este maior grau de sofisticação implicará em uma configuração e controle mais complexos é o que será visto no sub-item seguinte.

4.2. Controle de Monitores Remotos

Um monitor remoto (dispositivo RMON) pode ser configurado como uma função disponível em um sistema ou como um dispositivo dedicado. Configurado como dispositivo dedicado, o monitor é capaz de efetuar operações mais complexas.

A definição da MIB RMON contém características que suportam controle extensivo da estação de gerenciamento. Estas características dividem-se em duas categorias :

- Configuração

Tipicamente, um monitor remoto necessitará ser configurado para coletar dados. A configuração dita o tipo e a forma de dados para serem coletados. A MIB é organizada em grupos funcionais. Cada grupo terá uma ou mais tabelas de controle e uma ou mais tabelas de dados. Uma tabela de controle contém parâmetros que descrevem o dado na tabela de dados, que é somente para leitura. Assim, a estação gerente envia os parâmetros apropriados para configurar o monitor remoto para coletar os dados desejados. Os parâmetros são configurados pela adição de um novo registro na tabela ou alterando uma existente. Desse modo, funções para serem executadas pelo monitor são definidas e implementadas na tabela. Por exemplo, uma tabela controle pode conter objetos que especifiquem a origem dos dados coletados, tipos de dados.

Para modificar qualquer parâmetro na tabela de controle é necessário primeiro invalidar a entrada. Isto causa a retirada daquela entrada e de todos os registros associados em tabelas de dados. A estação gerente pode então criar um novo registro controle com os parâmetros modificados. O mesmo mecanismo é usado para apenas desabilitar uma coleção de dados. Quando um registro da tabela de controle é apagado, os registros das tabelas de dados associadas são apagados, e os recursos usados pelos registros são recuperados.

- Invocação de Ação

O SNMP providencia mecanismos não específicos para emitir um comando para um agente executar uma ação. SNMP tem apenas capacidade de ler valores de objetos e ajustar valores de objetos com visão MIB. Contudo, é possível usar o conjunto de operações SNMP para emitir um comando. Um objeto pode ser usado para representar

um comando, assim que uma ação específica é alcançada se o objeto é ajustado para um valor específico. Um número desses objetos são incluídos na MIB RMON. Em geral, estes objetos representam estados, e uma ação é executada se a estação gerente trocar o estado (pela troca do valor do objeto).

Percebe-se que o mecanismo de funcionamento do RMON é igual ao do SNMP, ou seja gerente e agente interagindo através de primitivas de comunicação, o que diferencia são as organizações das bases MIB e a maior autonomia do agente RMON. No próximo sub-capítulo veremos que o RMON já tem duas versões de bases MIB.

4.3. RMON1 e RMON2

Dois padrões básicos de protocolo RMON são especificados: RMON1 e RMON2, funcionalmente complementares.

Os objetos gerenciados do RMON1 operam somente na camada "*Media Access Control"* (MAC) e, segundo [Carvalho97], "oferece recursos ao administrador da rede para monitorar o tráfego e coletar informações estatísticas da operação de um segmento de rede local, além de realizar o diagnóstico remoto de falhas e erros ocorridos no segmento de rede a partir de funcionalidades de um analisador de protocolo suportadas pelo correspondente elemento RMON."

Porém, segundo [NetScout96], o fato do RMON1 só trabalhar na camada MAC, significa que este somente apresenta estatísticas para tráfego agregado - porém não apresenta estatísticas para camadas diferentes de várias pilhas de protocolos (ex. IP, FTP, IPX). Isto também significa que, por não serem capazes de monitorar a camada de rede, os dispositivos RMON1 não distinguem o tráfego neste segmento originado através de um roteador, o que é uma grande deficiência. Assim, muitas aplicações usuais como uma medição do tempo de resposta cliente/servidor ou uma provisão de estatística para as sete camadas, não é possível através deste protocolo unicamente.

Os objetos gerenciados do RMON2, por sua vez, operam no nível da camada de rede e camadas superiores, complementando portanto o RMON1, possibilitando coletar

informações estatísticas e monitorar a comunicação fim-a-fim e o tráfego gerado por diferentes tipos de aplicação. A figura 5 ilustra esta diferença.

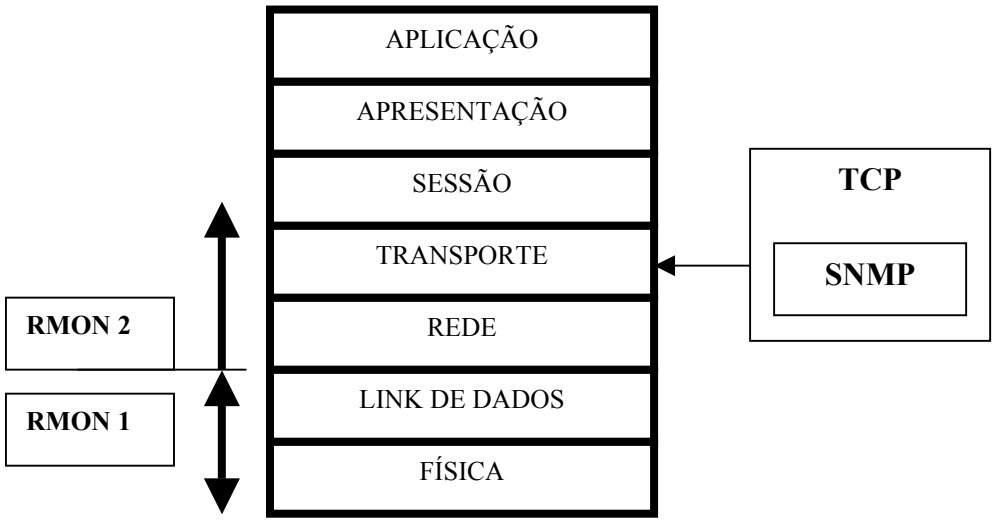


Figura 5 : Diferenças na Atuação da RMON1 e 2

Neste capítulo ficou claro que o protocolo SNMP atua na camada de aplicação do protocolo TCP/IP, que por sua vez atua na camada de transporte do modelo OSI.
A diferença maior entre a versão 1 e 2 do protocolo RMON, está nas camadas do modelo OSI onde atuam seus objetos gerenciados (variáveis), o que torna a versão 2 mais abrangente. Na próxima seção são abordadas em maiores detalhes os objetos gerenciados do RMON.

4.4 Bases de Dados do RMON.

A implementação das funções do protocolo RMON somente é viável mediante o suporte de uma base de dados de gerenciamento, a RMON-MIB, associada a cada elemento RMON da rede.

4.4.1. Grupos da RMON- MIB 1

Para a RMON1-MIB, foram especificados dez grupos básicos de variáveis, segundo RFC 2819, fornecendo várias opções de características de gerenciamento [Carvalho97] :

- Estatístico: mantém estatísticas de utilização, tráfego e taxas de erros ocorridos em um segmento de rede;
- Histórico: permite controlar o processo de amostragem (definição dos intervalos de amostragem) de informações do grupo estatístico e registrar tais informações, empregadas na análise do comportamento de uma rede e que oferecem subsídios para um gerenciamento pró-ativo;
- Alarmes: possibilitam estabelecer condições limites de operação de uma rede que deve provocar a geração de alarmes;
- Hosts: contêm informações relativas ao tráfego gerado e recebido pelos "hosts" conectados através da referida rede;
- Classificação de *n hosts* (*top n hosts*): permite classificar os "*hosts"* segundo critérios pré-definidos como, por exemplo, determinar quais os "*hosts"* conectados através da rede que geram maior tráfego em um dado período do dia;
- Matriz: contém informações de utilização da rede e taxa de erros na forma de matriz, associando pares de endereços MAC de elementos de rede;
- Filtro: define condições associadas a pacotes trafegados pela rede, que uma vez satisfeitas implicam captura de tais pacotes pelo elemento RMON ou no registro de estatísticas baseadas nos mesmos;
- Captura de Pacotes - determina como devem ser capturados os dados dos pacotes trafegados pela rede, a serem enviados aos gerentes. Como "default", são capturados os cem primeiros *bytes* dos pacotes filtrados pelo elemento RMON;

- Evento - define todos os eventos que implicam a criação de registros (“*logs”*) de eventos e o envio de informações pertinentes do elemento RMON aos gerentes.

A implementação de todos os grupos é opcional, embora exista uma relação de dependência entre alguns deles, como é o caso do grupo de "classificação de *n hosts*" em relação ao grupo de hosts, que se utiliza deste para para classificar quais os hosts que geraram maior tráfego.

4.4.2. Grupos da RMON-MIB 2

Para a RMON2-MIB, foram especificados, também, dez grupos básicos de variáveis, conforme RFC 2021, que incluem [Carvalho97]:

- Diretório de Protocolo; especifica uma lista dos protocolos - de rede, transporte e de camadas superiores - que o elemento RMON tem a capacidade de monitorar, sendo possível incluir, remover ou configurar entradas dessa lista;
- Distribuição de Protocolo: contém informações relativas ao número de *bytes* ou pacotes referentes a diferentes protocolos transferidos através de um determinado segmento de rede;
- Mapeamento de Endereços: relaciona endereços MAC e endereços de rede - ou endereços IP;
- Camada de Rede do *Host* - contabiliza o tráfego gerado e recebido por um “*host”*, cujo endereço de rede é conhecido pelo RMON;
- Matriz da Camada de Rede: contabiliza o tráfego existente entre um par de *hosts*, cujos endereços de rede são conhecidos pelo RMON;
- Camada de Aplicação do *Host:* contabiliza o tráfego, relativo a determinado protocolo, gerado e recebido por um “*host”*, cujo endereço de rede é conhecido pelo RMON;

- Matriz da Camada de Aplicação: contabiliza o tráfego, relativo a um determinado protocolo, existente entre um par de "*hosts"*, cujos endereços de rede são conhecidos pelo RMON;
- Histórico do Usuário: contém informações específicas de um usuário relativo ao tráfego gerado, período e intervalos de amostragem, entre outras informações;
- Configuração do Probe: contém a configuração dos parâmetros de operação do RMON;
- Conformidade RMON: define os requisitos de conformidade da RMON-MIB.

Para este trabalho é indiferente a versão de MIB que seja escolhida. A intenção é demonstrar como criar uma ferramenta de gerência e desenvolver uma interface com o agente.

O próximo Capítulo descreverá algumas ferramentas de gerência, citando suas características.

Capítulo 5 – Ferramentas Gerenciadoras

Existem várias opções de ferramentas de gerência, basta acessar a internet, utilizar uma página de busca, e procurar por gerência de redes. Há ferramentas de vários níveis, desde um simples "sniffer", que de certa forma poderia estar coletando dados da rede e

fornecendo subsídios para um administrador tomar uma decisão, até ferramentas altamente sofisticadas, com muitas informações com excelente visual. Independentemente de sua qualidade todas são ferramentas que a nível de modelo OSI se encontram na camada de aplicação.

Este capítulo visa demonstrar algumas funcionalidades de ferramentas existentes para a gerência de redes. Foram escolhidas aquelas mais conhecidas no ambiente de gerência, e duas outras: Olho Vivo e SAGRES por mais se aproximarem de nossa proposta para este trabalho, pois também utilizam Sistema Especialista.

5.1. Olho Vivo

O objetivo do Olho Vivo [SAENZ 96] é fazer uma gerência pró-ativa utilizando monitores remotos e sistemas especialistas. Ele é composto de monitor remoto RMON-MIB e um conjunto de regras de produção, consideradas o "centro nervoso" do sistema.

Ferramenta que funciona como um vigilante da rede, observando os indicadores de degradação do desempenho da rede através dos conceitos de agentes remotos (RMON-MIB) e sistemas especialistas e buscando soluções para os problemas encontrados.

Composta pelo agente RMON, módulo inteligente, um gerente e uma interface. O agente RMON utilizado foi o btng (Beholder The Next Generation) [OTTEN 93]. Esse monitor foi desenvolvido pelo grupo de pesquisa DNPAP (*Data Network Performance Analysis Project*) da Universidade Tecnológica de Delft, na Holanda, ele implementa a RMON-MIB completa. O beholder é um conjunto de coletores, onde cada um é responsável por um grupo de objetos da RMON MIB.

O módulo inteligente é o principal componente do Olho Vivo, analisando os resultados coletados pelo monitor (beholder) e sugerindo soluções ao administrador sobre os problemas de desempenho da rede.

Foram implementadas nove regras, cada uma referindo-se a problemas específicos da rede como:

Regra 1: referente ao número de pacotes descartados pelo monitor.

Regra 2: referente ao nível de “broadcast”, devendo este nível ser menor que 8%.

Regra 3: referente ao nível de multicast.

Regra 4: erros de alinhamento ou CRC, a taxa de erros deste tipo deve ser inferior a 2% do tráfego total.

Regra 5:, 6 e 7: avaliam contadores de pacotes com erro de comprimento.

Regra 8: taxa de colisões

Regra 9: taxa de utilização, não deve ultrapassar 40%.

O gerente utilizado foi o software SNM (SunNet Manager da Sun MicroSystems) e a interface com o administrador é o correio eletrônico. Quando a regra é verdadeira pela primeira vez envia um e-mail indicando o problema e o nome do arquivo de “log”.

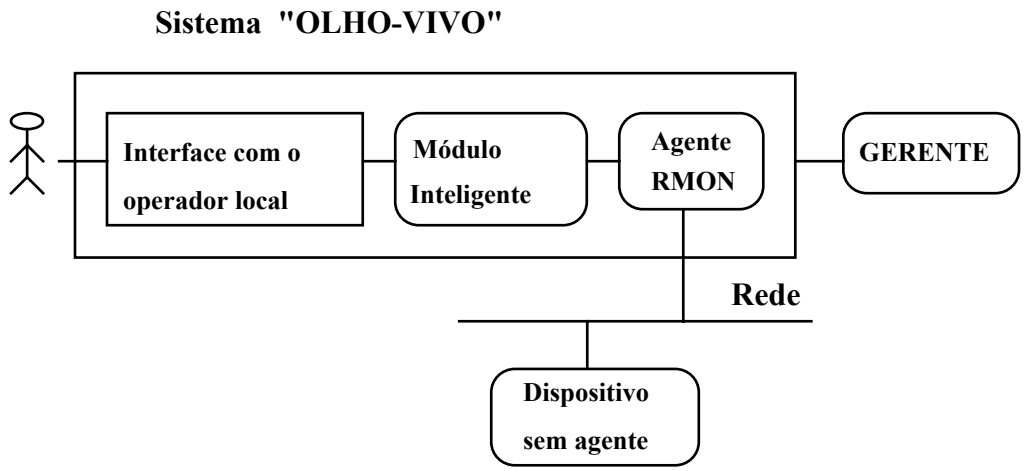


Figura 6: Diagrama de Blocos do Olho Vivo

Da figura 6 concluímos que o agente RMON, atuando remotamente, pois perceba o dispositivo sem o agente local, coleta informações da rede, repassa para o sistema especialista, que poderá disparar alguma regra da base de conhecimento, comunicando o

problema ao gerente, que enviará um e-mail para o operador através da interface que possui com este.

Uma das limitações do sistema é em relação à interface do usuário, é feita através de correio eletrônico, o ideal seria que essa interface fosse um "*shell"* para sistemas especialistas, de modo que seja possível agregar conhecimento à base de conhecimento de uma forma automática. Outro fator limitante é que o operador não interage com o sistema no sentido de alterar alguma variável de controle.
O sistema Olho Vivo serviu de referência para este trabalho, ambos tem similaridades no tocante a utilizarem Sistemas Especialistas e protocolo SNMP. Porém as deficiências apontadas acima formam as diferenças, pois houve preocupação neste trabalho com a interface com o usuário via o Sistema Especialista, bem como a questão de ser uma "Shell".

5.2. SAGRES

O SAGRES [HOLANDA 98] é uma ferramenta baseada em conhecimento projetada para a área de gerência de sistemas. Possibilita a coleta, a análise, o tratamento e a geração de ações relacionadas ao comportamento da rede, como conseqüência de regras previamente cadastradas em uma base de conhecimento, para tanto foram utilizados os conceitos de SGRBCs (Sistemas de Gerenciamento de Redes Baseado em Conhecimento). Esse sistema segue o modelo de gerência de redes adotado na internet, o SNMP v2.

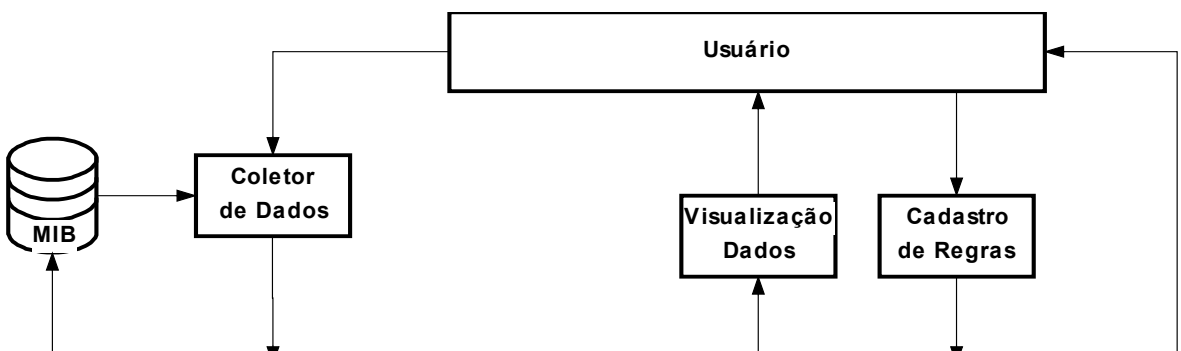


Figura 7: Diagrama de Blocos SAGRES.

Da figura 7 são identificados nove componentes. A descrição de cada um de seus componentes é relacionada a seguir:

- usuário: pode ser usuário ou administrador do sistema. Necessita de uma conta e senha para acessar as funcionalidades do sistema. O administrador deve fazer a manutenção de todas as contas do sistema;
- MIB: repositório de dados contendo todas as informações dos objetos gerenciados da rede;
- Coletor de Dados: realiza a coleta dos dados da rede e os armazena em uma base de dados para posterior visualização em forma textual ou gráfica. Este módulo coleta dados da MIB utilizando comandos GET;
- Tratamento dos Dados: é alimentado pelo módulo Coletor de dados e os dados são tratados antes de serem disponibilizados ao administrador ou usuário em forma de informação;
- Visualização de Dados: é alimentado pelo módulo de Tratamento de dados;

- Cadastro de Regras: é responsável pela interação entre administrador e Base de Conhecimento (BC), é através deste módulo que as inclusões, alterações e remoções de ações e regras na BC são realizadas;
- BC (Base de Conhecimento): é o repositório das ações e regras do sistema;
- Analisador de Regras: utiliza os dados coletados, ações e regras cadastradas para detectar e corrigir problemas no comportamento da rede;
- Ação na MIB: faz alterações nos parâmetros da rede mediante o processo de inferência sobre as regras do módulo analisador de regras.

A aplicação SAGRES é implementada parte em ambiente UNIX e parte em PC. A linguagem Delphi foi utilizada para o ambiente PC e a linguagem Tcl/Tk (Tcl versão 7.6 e Tk versão 4.2) e a ferramenta Scotty (versão 2.1.5) foram utilizadas para o ambiente UNIX.

5.3. Cisco Works

O CiscoWorks [CiscoWorks] é um conjunto de aplicações de gerência de rede baseado em SNMP. Aplicações Cisco Works são integradas a vários sistemas de gerência de redes como: SunNet Manager, Site Manager, Domain Manager e Enterprise Manager para Solaris; HP Open View para sistemas HP ou Sun e IBM Net View para AIX, que permitem monitorização do estado de dispositivos, facilidades de manutenção e resolução de problemas de dispositivos Cisco.

A seguir, algumas aplicações presentes no Cisco Works:

- Gerente de auto-instalação: permite instalar remotamente um roteador novo utilizando um roteador vizinho;

- Cisco View: mostra graficamente uma visão dos dispositivos físicos Cisco e informações como estado dinâmico, estatísticas e de configuração dos switches, roteadores, hubs, sevidores de acesso, concentradores e adaptadores Cisco. Oferece também funções de monitorização e resolução de problemas;
- Gerência de arquivos: cria e mantém um banco de dados que armazena um completo inventário da rede: hardware, software, versões de componentes operacionais, responsáveis por manutenção de dispositivos e locais associados;
- Facilidade de comando global: informações similares compartilhadas como senhas e listas de acesso podem ser configuradas para serem aplicadas automaticamente para um grupo comum de dispositivos;
- Monitor de saúde: fornece informações sobre estado de dispositivos, incluindo "buffers", carga de CPU, memóira disponível e protocolos/interfaces sendo utilizadas;
- Análise "offline" da rede: coleta dados históricos para análise "offline" de tendências de desempenho e padrões de tráfego. Com o SQL do Sybase projetado para variáveis SNMP é possível fazer consultas e gerar gráficos;
- Path Tool: permite análise de rota entre dois dispositivos, coletando dados de utilização e erros;
- Security Manager: configura procedimentos de segurança para aplicações e dispositivos de rede selecionados contra acessos não autorizdos, protegendo o ambiente Cisco Works através de pedido de login/senha;
- Software Manager: minimiza o custo de atualização, permitindo aos administradores centralizar a distribuição e gerência de software de roteador através da rede.

O Cisco Works é uma ferramenta proprietária, ou seja, utilizada em redes onde existam elementos tais como roteadores, "switches" etc, da plataforma Cisco. Tende a ser completa e dando todas as condições do administrador obter as informações que necessita deste elemento de rede.

5.4. Open View

O conjunto de produtos HP Open View [Openview]: HP Open View Professional Suíte, HP Open View Network Node Manager para Windows NT e o HP Open View Network Node Manager para UNIX compartilham um objetivo comum que é fornecer um conjunto de serviços essenciais que forneçam as bases para uma gerência de ambientes com múltiplos fabricantes, distribuído e de rede. Fornecendo serviços de:

- Interface de usuário: a interface do usuário é o mapa, por ele tem-se uma visão hierárquica da rede, representando suas subredes e elementos básicos, sendo estes elementos representados por ícones que mudam de cor conforme o estado do elemento na rede;
- Gerência de evento: notificam as aplicações que eventos específicos ocorreram e que alguma ação deve ser tomada;
- Descobridor: o descobrimento da rede é a checagem dos dispositivos da mesma e posterior representação dos elementos descobertos no mapa por ícones;
- Banco de Dados de gerência: é o repositório central onde os dados sobre a rede são armazenados, podendo ser um banco de dados proprietário ou não. Estes dados serão utilizados para relatórios, análise e plotagem de gráficos;
- Infraestrutura de comunicações: para a descoberta dos dispositivos é necessária uma infraestrutura de comunicação por duas razões: existência de múltiplos protocolos e facilidade de conexão entre aplicações e objetos gerenciados, por isto o uso do SNMP;
- Serviços integrados: as API's (Application Programming Interfaces) fazem parte dos serviços de integração do HP Open View e servem para integrar as aplicações entre si;
- Gerência de nó: as soluções de gerência de rede devem ser capazes de descobrir e dimensionar em mapas gráficos os dispositivos de rede

(roteadores, bridges, switches, servidores, desktops, impressoras) e nós. Estes elementos são geralmente numerosos. Os alarmes devem acionar o administrador assim que algum evento relevante aconteça com algum dispositivo ou nó.

A pesar do Open View ser uma ferramenta proprietária, ele se preocupa com a integração da diversas tecnologias de redes de fabricantes diferentes. Isto é importante nas redes atuais pois a diversidade de fabricantes envolvidos em uma configuração de rede geralmente é grande. Vejamos a seguir mais uma ferramenta proprietária.

5.5. Tivoli Enterprise

O Tivoli Enterprise [Tivoli], ferramenta da IBM, fornece soluções de administração de sistemas para pequenas e grandes organizações, centralizando a gerência dos recursos computacionais. A base da administração de sistemas empresariais Tivoli é o Tivoli Management Framework, com um banco de dados orientado a objetos, interface gráfica e serviços básicos utilizaods por outros produtos.

A plataforma de gerência Tivoli é baseada na arquitetura ORB (Object Rquest Broker) e utiliza completamente a especificação de grupo de gerência de objetos CORBA 1.1 em cada máquina onde o ambiente de gerência Tivoli está instalado roda um processo ORB chamado “oserv”. O “oserv” é um processo de longa duração que atua como um agente na máquina. Recebe todos os pedidos locais através da interface gráfica do usuário e interface de linha de comando (CLI), comunica-se com outros “oserv” de outras máquinas e inicia qualquer processo/método necessário na máquina local.

O Tivoli permite gerenciar diversas arquiteturas e variados sistemas operacionais UNIX e de PC como DOS, OS/2, Windows, Windows 95 e Windows NT. E possui facilidades básicas como programação de tarefas, notícias/mensagens de eventos, níveis de usuários/administradores dividindo as ações por permissões de uso (superior, sênior, admin e user), interface gráfica, CLI. Utiliza um banco de dados relacional para armazenar os dados e possui ferramentas para backup destes dados.

Possui uma linha de produtos em áreas como segurança, armazenamento, controle remoto, administração de mainframe, entre outros, que são instalados sobre esta base principal, Tivoli Management Framework.

Este capítulo forneceu uma visão de algumas ferramentas utilizadas na redes atuais em qualquer empresa, independente do tamanho. Duas em especial estudamos com mais interesse, Olho Vivo e Sagres, pois ambas utilizam a idéia do sistema especialista que é o ponto principal deste trabalho. Os capítulos 2 e 3 trataram sobre gerência de redes, no capítulo 6 mudaremos totalmente de assunto passando para a inteligência artificial, mais especificamente os sistemas especialistas.

Capítulo 6 – Sistemas Especialistas

"Software inteligente que utiliza conhecimentos e um processo de inferência para resolver problemas, difíceis e que requerem um especialista humano para a sua solução. Os conhecimentos de um sistema especialista são compostos de fatos e heurísticas. " [Feigenbaum79].

6.1. Introdução

Os Sistemas Especialistas de primeira geração nasceram na década de 60. O objetivo da época era desenvolver um programa que pudesse "pensar". Em 1965 surgiu o DENDRAL, que se restringiu ao meio acadêmico [Lindsay80]. Em 1975 surgiu o MYCIN, foi o primeiro a utilizar fator de certeza [Buchanan84]. Em 1982 com o surgimento do XCON os sistemas especialistas saíram do meio acadêmico e chegaram

até as industrias. Em 1990 o SE's tiveram um grande avanço, pois passaram a utilizar a lógica difusa e não mais a lógica clássica procedural.

A definição de Feigenbaum [Feigenbaum79] é de um sistema especialista de primeira geração.
Já sistemas com :

- Aquisição automatizada de conhecimentos.
- Integração de múltiplos métodos de "raciocínio".

São chamados de sistemas especialistas de segunda geração.
As linguagens para o desenvolvimento de sistemas especialistas podem ser: Prolog, Lisp etc.

Podemos fazer comparações entre a programação convencional e os Sistemas Especialistas:

Programação Convencional	Sistema Especialista
1. Numérico	Simbólico
2. Algorítmico	Heurístico
3. Informação e controle integrados	Conhecimento separado do Controle
4. Dificuldade de modificação	Fácil alteração
5. Informação precisa	Informação incerta
6. Interface com comandos	Diálogo natural com explicações
7. Um resultado final é obtido	Uma recomendação é explicada
8. Solução ótima	Solução aceitável
9. Comandos seqüenciais	Dirigidos pela máquina de inferência
10. Projeto estruturado	Pouca ou nenhuma estruturação
11. Sem explanação	Facilidade em explicar o raciocínio
12. Crescimento em "módulos"	Crescimento incremental
13. Buscas são restritas ou ausentes	Constantemente sobre as regras

Tabela 1: Comparativo entre a Programação Convencional e SE.

A seguir vamos analisar cada um dos treze itens expostos na tabela acima e tecer um comentário para melhor entendimento:

1. Numérico "2+3=5" é um resultado aceito pela matemática, já no plano simbólico "A≡B" significa que A é logicamente equivalente a B;

2. O sistema especialista raciocina heuristicamente, se aproximando do raciocínio humano, onde o conhecimento informal, a prática e experiência são importantes, ou seja não são baseados apenas em algoritmos rígidos que buscam apenas a lógica;

3. Diferentemente dos programas convencionais que mesclam lógica e conhecimento, os sistemas especialistas separam a base de conhecimento da inferência;

4. A modificação de um programa convencional pode ser extremamente trabalhosa dependendo da complexidade do programa, já que lógica e conhecimento se misturam, isto não ocorre no sistema especialista, pois separa as duas coisas;

5. O sistema especialista trata bem as informações do tipo “é provável”, “pode ser que” etc;

6. A lógica procedural entende apenas comandos explícitos, já os SE’s trabalham com palavras definidas por um alfabeto;

7. Os sistemas especialistas sugerem uma solução para um problema e podem também explicar como chegaram a esta sugestão;

8. O sistema procedural gera uma resposta fidedigna ao algoritmo, já os SE’s emitem uma sugestão aceitável de acordo com o conhecimento que detêm.

9. A programação convencional segue a estruturação do algoritmo, os SE’s seguem o “raciocínio” da máquina de inferência;

10. A programação atual é dita estruturada, os SE’s não trabalham com estruturação, buscam similaridade ao raciocínio humano;

11. Os SE's tem a possibilidade de demonstrar o raciocínio utilizado para chegar à sugestão de solução;

12. A programação convencional cresce geralmente com módulos novos ou funções novas, já os SE's podem crescer apenas pela adição de uma regra nova à base de conhecimento;

13. Os SE's trabalham basicamente em constantes buscas na base de conhecimento de regras que se encaixem aos fatos sugeridos para o problema.

Neste trabalho foi desenvolvido um sistema especialista, porém convém comentar que na inteligência artificial existe também uma tecnologia chamada "Raciocínios Baseados em Casos" (RBC's).

O Raciocínio Baseado em Casos pode ser usado para várias tarefas de raciocínio como por exemplo: propor soluções para novos problemas, antecipar, evitar e explicar fracassos em soluções propostas, e adequar e/ou reparar soluções propostas. Tem também uma grande capacidade de aprendizado.

Porém pelo tipo de problema que este trabalho visa a responder, a opção foi por Sistemas Especialistas, devido a força destes em relação a base de conhecimento, ou seja ao domínio de uma área de conhecimento e a forte capacidade incremental.

Normalmente um sistema especialista é criado para resolver problemas em um determinado domínio, área de interesse específico para as quais podemos desenhar um sistema de IA, cujo conhecimento utilizado é fornecido por pessoas que são especialistas naquele domínio.

Programas de computador que tentam resolver problemas que os seres humanos resolveriam emulando o raciocínio de um especialista, aplicando conhecimentos

específicos e inferências também fornece uma boa idéia do que são sistemas especialistas.

Os Sistemas Especialistas são diferentes das aplicações típicas por causa de sua arquitetura. Um dos princípios fundamentais no projeto de sistemas especialistas é a separação do conhecimento de domínio (por exemplo, áreas específicas da medicina ou geologia) dos programas que "raciocinam" com este conhecimento [Buchanan89]. Portanto, existe uma distinta divisão entre o componente de conhecimento do sistema e o componente de raciocínio ou máquina de inferência. A máquina de inferência é bem generalizada e usualmente poderá trabalhar com diferentes conjuntos de conhecimento [Peper91].

Um modelo básico da arquitetura dos sistemas especialistas pode ser apresentado como na figura 8, com três componentes básicos: a base de conhecimento, a máquina de inferência, e a interface com usuário.

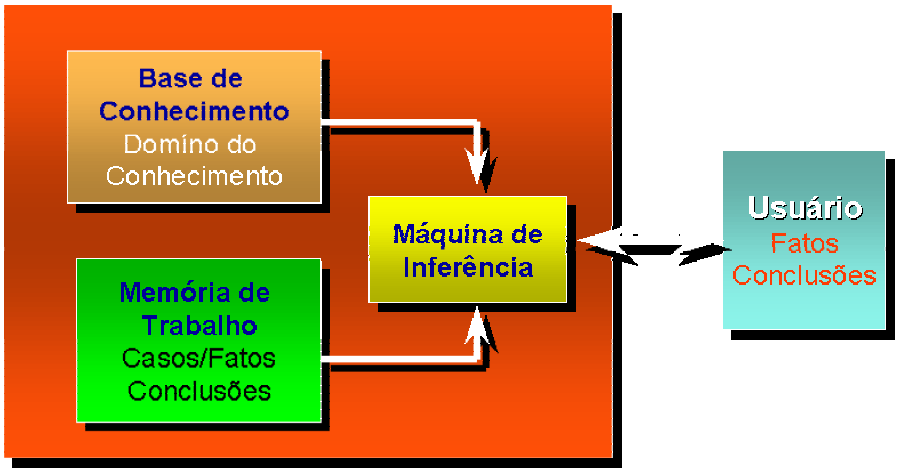


Figura 8: Estrutura Convencional de um Sistema Especialista

Um sistema convencional é baseado em um algoritmo, emite um resultado final correto e processa um volume de dados de maneira repetitiva enquanto que um sistema especialista é baseado em uma busca heurística e trabalha com problemas para os quais não existe uma solução convencional organizada de forma algoritmica disponível ou é muito demorada.

Um Sistema Especialista é aquele que é projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão, apoiado em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano.

Para tomar uma decisão sobre um determinado assunto, um especialista o faz a partir de fatos que encontra e de hipóteses que formula, buscando em sua memória um conhecimento prévio armazenado durante anos, no período de sua formação e no decorrer de sua vida profissional, sobre esses fatos e hipóteses. E o faz de acordo com a sua experiência, isto é, com o seu conhecimento acumulado sobre o assunto e, com esses fato e hipóteses, emite a decisão.

Durante o processo de raciocínio, vai verificando qual a importância dos fatos que encontra comparando-os com as informações já contidas no seu conhecimento acumulado sobre esses fatos e hipóteses. Neste processo, vai formulando novas hipóteses e verificando novos fatos; e esses novos fatos vão influenciar no processo de raciocínio. Este raciocínio é sempre baseado no conhecimento prévio acumulado. Um especialista com esse processo de raciocínio pode não chegar a uma decisão se os fatos de que dispõe para aplicar o seu conhecimento prévio não forem suficientes. Pode, por este motivo, inclusive chegar a uma conclusão errada; mas este erro é justificado em função dos fatos que encontrou e do seu conhecimento acumulado previamente.
Um sistema especialista deve, além de inferir conclusões, ter capacidade de aprender novos conhecimentos e, desse modo, melhorar o seu desempenho de raciocínio, e a qualidade de suas decisões.

A forma de representação do conhecimento e ainda paradigma de programação de Sistemas Especialistas é a que envolve regras de produção "*production rules*". Essas regras são simplesmente um conjunto de condições no estilo SE... ENTÃO..., com a possibilidade de inclusão de conectivos lógicos relacionando os atributos no escopo do conhecimento e o uso de probabilidades, como vemos no exemplo a seguir:

SE carne = vermelha
E cor preferida = tinto
OU temperatura = ambiente
ENTÃO
melhor vinho = exemplo CNF 70;

Um sistema de bom tamanho tem em torno de uma centena de regras (considerando aqueles que utilizam regras). Visando uma maior viabilidade econômica na implementação de um sistema especialista, e considerando-se que diversos sistemas compartilham uma *máquina de inferência* e outras características comuns de ambiente, foram criadas ferramentas, "*shells"*, aptas a realizar muito do trabalho necessário para transpor um sistema especialista para um computador. Essas ferramentas permitem que o criador do sistema preocupe-se somente com a representação do conhecimento dos especialistas, deixando para a "*Shell"* a tarefa de interpretar o conhecimento representado e executá-lo em uma máquina, além de permitir depurações e explicações de como o computador chegou àquela(s) conclusão(ões). A principal função de uma "*Shell"* é simplificar ao máximo o trabalho de implementação de um sistema especialista, através de uma arquitetura padrão, onde apenas o conhecimento é o elemento variável, podendo ser utilizado para várias áreas do conheciemnto. E predominantemente permite o uso por qualquer pessoa sem conhecimentos de informática.

De um modo geral, sempre que um problema não pode ser algoritmizado, ou sua solução conduza a um processamento muito demorado, os Sistemas Especialistas podem ter uma saída, pois possuem o seu mecanismo apoiado em processos heurísticos.

Em suma Sistemas Especialistas são programas de computador que procuram atingir soluções de determinados problemas do mesmo modo que especialistas humanos, se estiverem sob as mesmas condições. Apesar das limitações das máquinas, é possível, hoje, a construção de Sistemas Especialistas com alto grau de desempenho, dependendo da complexidade de sua estrutura e do grau de abrangência desejado.

6.2. Arquitetura de Sistemas Especialistas

Nesta seção são analisados os componentes de uma arquitetura padrão de Sistema Especialita.

6.2.1 Base de Conhecimento

A marca principal de um Sistema Especialista é o uso do conhecimento específico de seu domínio de aplicação através de um programa de raciocínio relativamente simples. Neste sentido, o termo "base de conhecimento" é utilizado para significar a coleção de conhecimento do domínio, ou seja, as informações, a nível de especialista, necessárias para resolver problemas de um domínio específico. Portanto, este conhecimento precisa ser organizado de uma maneira adequada para que a máquina de inferência consiga tratá-lo convenientemente. O conhecimento em um sistema especialista consiste de *fatos* e *heurísticas*. Os fatos constituem as informações que estarão sempre disponíveis para serem compartilhadas e atualizadas pelo especialista do domínio. As heurísticas são regras práticas que caracterizam o nível de tomada de decisão do especialista em um domínio. Portanto, uma base de conhecimento pode ser vista como um conjunto de regras, cada qual podendo ser validada independentemente de estrutura de controle.

Um dos problemas mais sérios, e ao mesmo tempo muito comum, encontrado na implementação de sistemas especialistas, é que usualmente parece impossível fornecer um conhecimento completo sobre o qual o sistema vai operar. Portanto, o nível de desempenho de um sistema especialista está relacionado ao tamanho e a qualidade de sua base de conhecimento.

6.2.2. Regras de Produção

A representação do conhecimento através de regras de produção é popular por possuir as seguintes vantagens:

- *Modularidade*: cada regra, por si mesma, pode ser considerada como uma peça de conhecimento independente;
- *Facilidade de edição* (uma conseqüência da modularidade): novas regras podem ser acrescentadas e antigas podem ser modificadas com relativa independência;
- *Transparência do sistema*: garante maior legibilidade da base de conhecimentos.

Portanto, é preciso ter em mente que a modularidade de um sistema baseado nessa arquitetura permite a construção passo-a-passo da base de conhecimentos, ou seja, é possível realizar vários testes com apenas um subconjunto de regras concluído. Obviamente, sabe-se que menos regras implicam geralmente em um menor número de casos abrangidos.

6.2.3. Máquina de Inferência

Segundo Minsky, "... o conhecimento é útil somente quando podemos explorá-lo para ajudar a alcançarmos nossos objetivos." [Minsky86]. Nos sistemas especialistas, a máquina de inferência cumpre este papel, representando o meio pelo qual o conhecimento é manipulado, utilizando-se das informações armazenadas na base de conhecimento, para resolver problemas. Para isto, deve haver uma linguagem ou um formato específico no qual o conhecimento possa ser expresso para permitir o "raciocínio" e inferência. Métodos de inferência são necessários para fazer uso apropriado e eficiente dos itens em uma base de conhecimento para alcançar alguns propósitos tal como o diagnóstico de doenças.

A máquina de inferência, de certo modo, tenta imitar os tipos de pensamento que o especialista humano emprega quando resolve um problema, ou seja, ele pode começar com uma conclusão e procurar uma evidência que a comprove, ou pode iniciar com uma evidência para chegar a uma conclusão. Em sistemas especialistas, estes dois métodos são chamados de *"backward chaining"* e *"forward chaining"* respectivamente. Nem

todos os sistemas utilizam a mesma abordagem para a representação do seu conhecimento, portanto, a máquina de inferência deve ser projetada para trabalhar com a representação de conhecimento específica utilizada.

6.2.4. Interface Com o Usuário

A interface com o usuário visa facilitar a comunicação entre o sistema especialista e o usuário. Permite a interação com o sistema através da entrada de fatos e dados e através da saída em forma de perguntas , conclusões e explicações.

Muitos princípios baseados nas teorias cognitivas têm sido propostos para projetos de interface, como resultado de pesquisas na área de interação homem-máquina. Uma das considerações principais no projeto de qual quer interface homem-máquina deve ser a facilidade de uso, reduzindo ao máximo a carga cognitiva sobre o usuário.

6.3. Conhecimento Procedural x Conhecimento Declarativo

Quando uma pessoa tem conhecimento de algum fato, certamente ela poderá extrair tudo o que souber sobre aquele fato quando bem entender. Porém, na Inteligência Artificial, existe um problema a mais quanto ao uso do conhecimento.

Supõe-se que o conhecimento por si só já é o suficiente para a resolução de problemas. Essa é a idéia por trás do *conhecimento declarativo*: não há preocupações quanto ao seu uso, somente quanto à sua posse e especificação, e ela já garantirá o alcance dos objetivos desejados.

Porém, a realidade é bem mais problemática. Um computador não é capaz de decidir qual a próxima informação que ele utilizará para o desenvolvimento de uma atividade. Faltam às máquinas um modo menos metódico e linear de ação. É necessário

que especifiquemos uma estratégia de uso do seu "saber". Mas, qual é o problema, quando a máquina possui o potencial necessário? Vejamos um exemplo:

Todos nós sabemos calcular o fatorial de um número. O fatorial de zero é um, e o fatorial dos demais números positivos é simplesmente ele multiplicado pelo fatorial do seu antecessor (como quando dizemos que o fatorial de 5 é 5 vezes o fatorial de 4). Ao colocarmos essa representação no computador, podemos representar simbolicamente por

Fatorial (n) = n x Fatorial(n - 1)
Ah, e é claro: Fatorial(0) = 1

O conhecimento está aí: agora deve-se ditar ao computador o meio pelo qual ele utilizará tais afirmativas. Uma estratégia bastante comum é simplesmente percorrer as informações na ordem: inicialmente, a primeira; depois, se necessário a seguinte, e assim por diante. Mas, se deseja-se calcular o prosaico fatorial de 2? Para a máquina, fatorial(2) = 2 x fatorial(1). E quanto é o fatorial de 1? Nesse caso, deve-se voltar à primeira informação de como se calcula o fatorial, Fatorial (n) = n x Fatorial(n - 1). Assim conclui-se que fatorial(1) é 1 vezes fatorial de zero. E o fatorial de zero? Observe que, como tem-se um mecanismo fixo de extração de dados, não é possível avaliar a segunda informação enquanto não terminada a primeira. Assim, simplesmente o computador calcularia que o fatorial de zero é zero vezes o fatorial de menos um! E continuaria eternamente com essa regra.

Esse tipo de conhecimento que depende de uma regra de extração chama-se de *conhecimento procedural*. Nota-se que, se simplesmente invertida a ordem das informações, chegaria-se a um resultado (sempre antes de calcular um fatorial de um número, a máquina verificaria a possibilidade deste número ser zero). Por isso, o uso do conhecimento é uma questão maior que simplesmente uma descrição do saber.

6.4. Métodos de Extração de Conhecimento

O modo mais comum de utilização de um sistema especialista é o encadeamento para trás. O projetista deve incluir na definição da base quais os atributos que devem ser encontrados (ou seja, os objetivos - *goals* - do sistema especialista). A máquina de inferência encarrega-se de encontrar uma atribuição para o atributo desejado nas conclusões das regras (após o ENTÃO...). Obviamente, para que a regra seja aprovada, suas premissas devem ser satisfeitas, obrigando à máquina a encontrar os atributos das premissas para que possam ser julgadas, acionando um encadeamento recursivo. Caso o atributo procurado não seja encontrado em nenhuma conclusão de regra, uma pergunta direta é feita ao usuário.

6.4.1. Encadeamento Progressivo ("Forward Chaining")

"*Estratégia de inferência que começa com um conjunto de fatos conhecidos, derivando novos fatos a partir de regras que casem com suas premissas e com fatos conhecidos, e este processo prossegue continuamente até chegar a uma meta (um objetivo), ou até que não exista mais nenhuma regra com premissas que casem com os novos fatos derivados".*

Uma vez que o sistema tenha um fato na memória de trabalho, a máquina de inferência varre a base de conhecimento em busca de uma regra que case com a premissa, quando um regra é encontrada, esta é levada à memória de trabalho e o ciclo recomeça a fim de verificar novas regras que casem com a premissa, nesta etapa as regras já acessadas são ignoradas, o processo prossegue até que todas as regras sejam varridas.
Vejamos um exemplo, sejam as regras abaixo:
Regra #1
IF a **AND** b **THEN** c com sintaxe de I A temos: a , b → c
Regra #2
IF d **THEN** e : d → e

Regra #3

IF f **AND** e **THEN** b : f , e → b

Regra #1: a , b → c
Regra #2: d → e
Regra #3: f , e → b

Tabela 2: Exemplo Base de Conhecimento

	MT inicial	Regra disparada	MT final
1º Ciclo	d, f, a	d → e	d, f, a, e
2º Ciclo	d, f, a, e	f, e → b	d, f, a, e, b
3º Ciclo	d, f, a, e, b	a,b → c	d, f, a, e, b, c

Tabela 3: Descrição dos Ciclos de uma MI com Encadeamento Para-frente.

É possível também que neste processo de extração do conhecimento ocorram situações de conflito, ou seja, mais de uma regra atende às condições de disparo. A seção abaixo irá abordar os meios pelos quais pode-se resolver estes conflitos.

6.4.1.1. Resolução de conflito

Em muitos casos pode ocorrer um conflito entre as regras a serem disparadas, conflito este que pode até chegar à contradição de uma regra pela outra. Nos sistemas que utilizam o conceito de resolução de conflito, a máquina de inferência segue um processo de três passos: reconheça-resolva-aute, dado um ciclo de varredura através da regras da base de conhecimento, seus objetivos são:

Reconheça: case as premissas de todas as regras para os fatos listados na memória de trabalho, e identifique aquelas regras que podem disparar;

Resolva: se mais de uma regra puder disparar, escolha uma regra que possa disparar de acordo com alguma estratégia;

Atue: dispare a regra e adicione sua conclusão na memória de trabalho.

Quanto as estratégias de resolução de conflito, as mais conhecidas são:
1. Primeira regra que case o conteúdo na memória de trabalho;
2. Regra de prioridade mais alta;
3. Regra mais específica
4. Regra que se refere ao elemento mais recente adicionado à memória de trabalho;
5. Não dispara uma regra que já foi disparada;
6. Dispara todas as regras com linhas distintas de raciocínio;
7. Uma escolha aleatória sobre as regras do conjunto conflito.

6.4.2. Encadeamento Regressivo ("Backward Chaining")

"Estratégia de inferência que tenta provar uma hipótese ou conclusão, combinando dados que suportem tal resposta".
Neste caso já suspeitamos da solução do problema, temos que tentar prová-la, validando os dados que geram tal hipótese.
A partir de uma meta na memória de trabalho, verifica-se se já não foi provada alguma vez, pois algum outro conhecimento poderia já ter provado esta meta. Se esta meta ainda não foi provada a base de conhecimento é varrida em busca de um casamento de regra, este casamento é verificado indo de encontro a lado posterior ao ENTÃO da regra.
A máquina de inferência verifica se as premissas da regra já estão na memória de trabalho, se não estiverem estas passam a ser as novas metas, que buscam novos casamentos de regras, este processo continua até que o sistema encontre uma premissa que não é suportada por nenhuma regra, esta meta é então chamada de *primitiva*.

6.4.3. Resumo

Este capítulo abordou os detalhes relevantes dos Sistemas Especialistas necessários para o entendimento da ferramenta que será desenvolvida. A opção pelo método de extração do conhecimento foi o "Backward Chaning", pois se adequa perfeitamente a situação de

partida através de uma hipótese, que adere a proposta deste trabalho. Nove regras foram criadas para simular uma base de conhecimento. No capítulo seguinte será explorado em detalhe os passos para construção da ferramenta, bem como o resultado de algumas simulações.

Capítulo 7 – Implementação

O objetivo fundamental deste trabalho é o desenvolvimento de uma ferramenta de gerência de redes. Apresenta diferencial destacado pelo fato de ser configurável pelo administrador, sem que este necessite ter qualquer conhecimento sobre inteligência artificial ou sistemas especialistas, exigindo apenas o conhecimento em sua área de especialidade.

Dois destaques a serem alcançados nas implementações:

- Desenvolver uma interface com o gerente SNMP, de tal forma que o administrador possa com facilidade, alterar as variáveis que foram previamente configuradas, excluindo ou inserindo novas variáveis de interesse para sua gerência.
- Sistema Especialista que proporcione facilidade nas operações de alteração, eliminação ou criação de regras novas.

Predomina neste trabalho a preocupação com o desenvolvimento de uma interface que possibilite ao administrador facilidade de incorporar seu conhecimento à ferramenta. O conteúdo inicial da base de conhecimento parte de nove regras, que pretendem servir de modelo a fim de facilitar o entendimento do funcionamento. De forma idêntica foram selecionadas as variáveis para monitoração. O objetivo deste trabalho não é desenvolver uma ferramenta completa para a gerência de rede, mas um meio para o administrador

utilizar seu conhecimento de forma sistêmica, de maneira a preservá-lo da necessidade de dedicação exclusiva à observação dos parâmetros de rede e da inferência pela busca do possível problema em função da distorção de valor de algum parâmetro.

7.1. Modelo Proposto

Na figura 9 abaixo fornece a idéia do contexto do modelo. A interface SNMP e o Sistema Especialista estão instalados no servidor da rede, máquina a qual o administrador utiliza para a gerência da rede. O agente RMON está em um host qualquer da rede, como também poderia estar no próprio servidor, ou em outro segmento da rede, ou mesmo em outra rede distribuída conectada a esta.

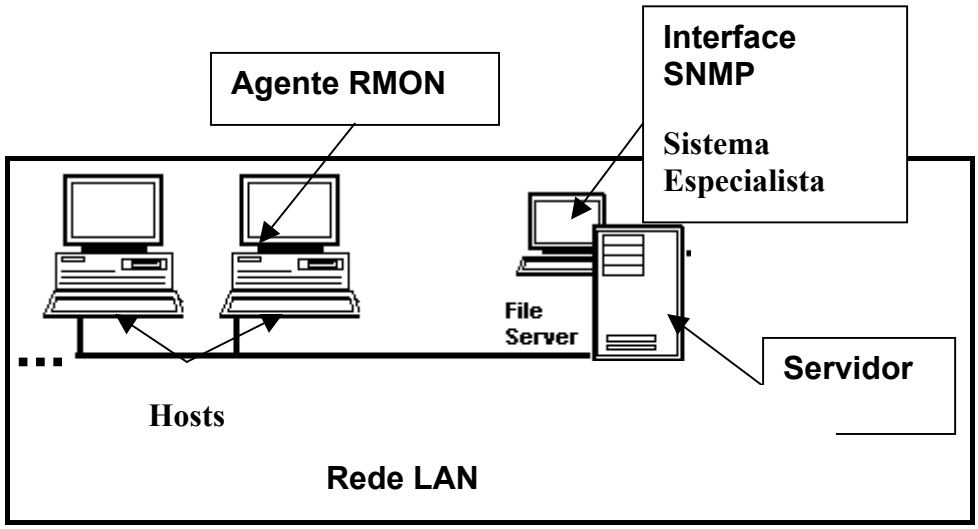
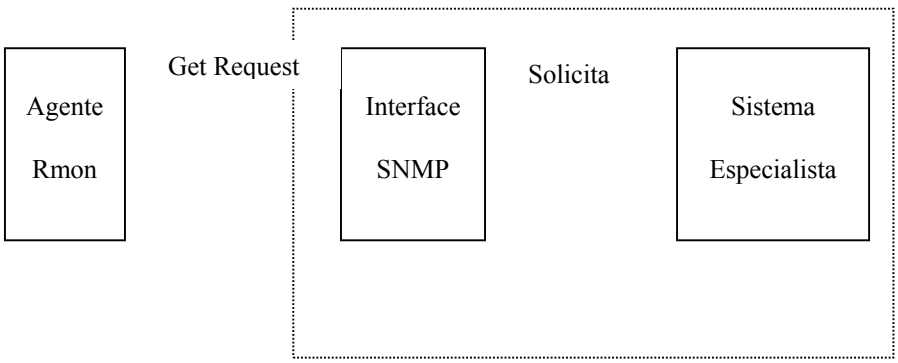


Figura 9: Diagrama de Contexto do Modelo Proposto.

O diagrama abaixo mostra a interação do Sistema Especialista da interface SNMP e do agente RMON:



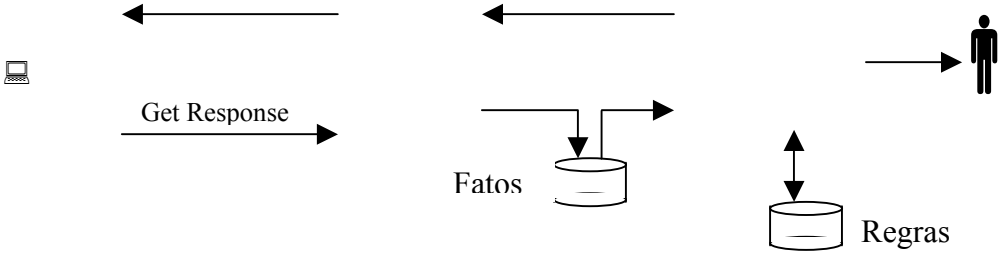


Figura 10: Diagrama de Blocos do Sistema

A interface SNMP e o Sistema Especialista foram desenvolvidos a fim de compor um sistema construído para o administrador de redes, a fim de que este possa interagir tanto com a interface SNMP como com o sistema especialista.

Atualmente a maioria dos elementos de redes já traz agentes RMON incorporados. Porém a interface SNMP é mais difícil encontrá-la isoladamente, quando encontrada não propicia a possibilidade de ajusta-la às necessidades específicas de cada sistema. Por esta razão foi imprescindível para este trabalho o desenvolvimento de uma interface SNMP, implementada na linguagem C [Microsoft], que irá através das primitivas do protocolo SNMP, gerar mensagens PDU do tipo GET-REQUEST, para buscar os valores das variáveis da RMON-MIB solicitadas pela interface SNMP. Foi utilizada a linguagem C pelas suas características favoráveis de portabilidade e disseminação.

A comunicação entre a interface SNMP e o agente se dará através das primitivas do protocolo SNMP, chamadas PDU's, basicamente será utilizada a solicitação, para o agente RMON, de uma lista atualizada das variáveis através da PDU get-request. O agente enviará em resposta a lista atualizada das variáveis solicitadas através da PDU get-response.

A comunicação entre a interface SNMP e o Sistema Especialista se dará através de arquivo texto, chamado de "fatos" na figura 10, no qual serão repassados para o sistema especialista, com atualizações em espaços de tempo pré-determinados, a relação de variáveis previamente selecionadas pelo administrador.

Na continuidade do propósito de oferecer uma ferramenta amigável para o administrador, foi desenvolvido um sistema especialista em Prolog [SWI], que permitirá ao administrador de rede, mesmo sem nenhum conhecimento de Inteligência Artificial, a alteração de variáveis para monitoração e a alteração, incrementando ou decrementando, das regras na base de conhecimento.

De posse das variáveis, da base de conhecimento e das regras o sistema especialista poderá iniciar suas inferências e apontar a provável causa de uma anormalidade.

7.2. Interface SNMP

Para o desenvolvimento desta interface, houve uma dificuldade para se encontrar uma plataforma aberta com a qual fosse possível uma interação com o agente SNMP. Há muitos "pacotes" disponíveis, porém não há a possibilidade de se acessar o agente para leitura dos dados da MIB. Foi utilizado como ambiente para simulação o sistema "ucd-snmp Version: 4.2.6" [Hardaker03]. Foi desenvolvido um programa na linguagem C a fim de acessar o agente deste sistema e ler as variáveis da MIB, a leitura da base é então direcionada para um arquivo texto, chamado fatos, em formato adequado para que o sistema especialista possa ler.

A figura 11 identifica o contexto onde está inserida a interface SNMP. Está instalada em um servidor em rede LAN, o qual utiliza sistema operacional Linux.

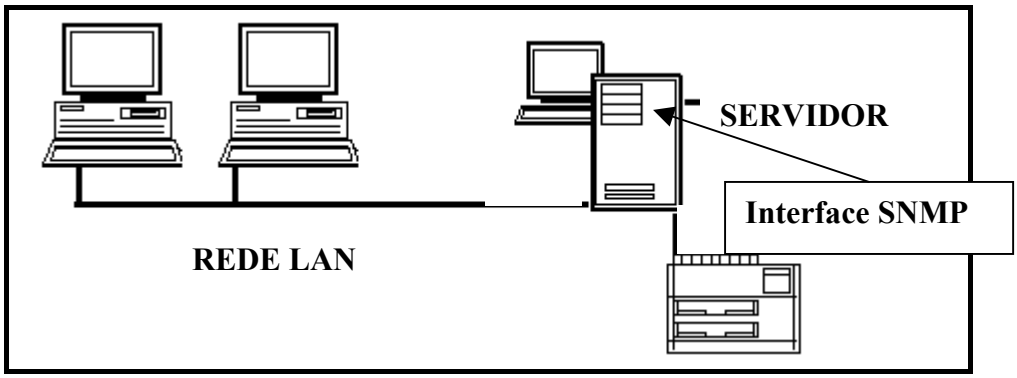


Figura 11: Contexto da Interface SNMP

Abaixo na figura 12 é mostrado o diagrama de blocos da Interface SNMP, detalhando as opções possíveis para o usuário:

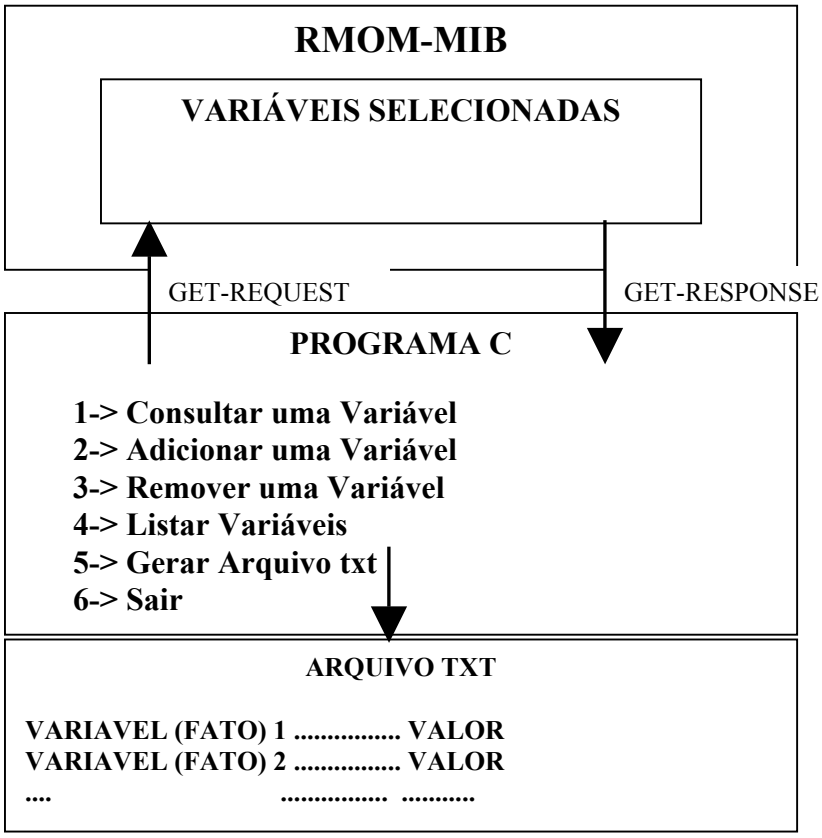


Figura 12: Diagrama de Blocos Interface SNMP

O elemento central da interface é o programa desenvolvido na linguagem C, o qual permitirá ao usuário a seleção de quais variáveis irá utilizar para a geração do arquivo texto, no anexo1 é apresentada a relação total de variáveis disponíveis na RMON-MIB. Também é possível ajustar o intervalo de amostragem, ou seja, o tempo decorrido entre as leituras da RMON-MIB e a atualização das variáveis no arquivo texto.

Uma vez tendo as variáveis definidas, haverá um script Crontab, via sistema operacional, que executará a prazos estabelecidos o programa em C, que buscará as atualizações destas variáveis nos agentes RMON, através de primitivas de comunicação SNMP, chamadas PDU's (Unidades de Dados de Protocolo), a mensagem PDU pode ser encarada como um objeto que contém variáveis que possuem nomes e valores. Através de uma PDU "Get-Request" encaminhada para o agente, pode-se solicitar os valores de uma lista de variáveis. Da mesma forma o agente responde a este pedido com uma resposta "Get-Response" informando os valores das variáveis solicitadas.

A interface SNMP de posse dos valores atualizados das variáveis disponibiliza um arquivo texto, chamado fatos, contendo as variáveis e seus valores. Estes valores são lidos pelo Sistema Especialista.

Para este trabalho foi utilizado para as simulações do programa Interface SNMP uma base MIB I, pois não foi encontrado sistema aberto, que permitisse a interação através de linguagem C, que apresentasse um agente RMON. A utilização da MIB I não implica em nenhuma alteração na interface SNMP a não ser do nome das variáveis selecionadas. O ambiente utilizado irá contemplar em breve um agente RMON, conforme informações contidas na documentação. No sub-capítulo 7.6 podem ser vistas as telas da interface nas simulações que foram executadas.

7.3 Variáveis Selecionadas

Para este trabalho foi selecionado um conjunto de nove variáveis, pelo simples fato de já terem sido escolhidas no sistema Olho Vivo [SAENZ 96], o limite de escolha de variáveis recai sobre a disponibilidade da base RMON-MIB. No presente trabalho estas variáveis têm apenas o caráter de demonstração para o funcionamento da Interface SNMP e Sistema Especialista. O administrador poderá selecionar as variáveis que julgar de interesse. São elas:

- **EtherHistoryDropEvents:** número de vezes que houve descarte de pacotes em função de falta de recursos do sistema.
- **EtherHistoryBroadcastPkts:** número de pacotes válidos recebidos durante o intervalo de amostra que foram direcionados para o endereço de broadcast.
- **EtherHistoryMulticastPkts:** número de pacotes válidos recebidos durante o intervalo de amostra que foram direcionados para o endereço de multicast.

- **EtherHistoryCRCAlignErrors:** número de pacotes recebidos durante o intervalo de amostragem, que tem o comprimento entre 64 e 1518 octetos inclusive, excluindo os bit's de frame, incluindo apenas os bit's FCS (Frame Check Sequence).
- **EtherHistoryUndersizePkts:** números de pacotes válidos recebidos durante o intervalo de amostragem, que tem o comprimento menor do que 64 octetos.
- **EtherHistoryOversizePkts:** número de pacotes válidos recebidos durante o intervalo de amostragem, que tem o comprimento maior do que 1518 octetos.
- **EtherHistoryFragments:** número total de pacotes recebidos durante o intervalo de amostragem, que tem comprimento inferior a 64 octetos.
- **EtherHistoryCollisions:** melhor estimativa durante o período de amostragem no segmento ethernet do número total de colisões.
- **EtherHistoryUtilization:** melhor estimativa durante o período de amostragem nesta interface da utilização da camada física de rede.

Características das variáveis:

Variável	Tipo	Acesso	Status
EtherHistoryDropEvents	Contador	Leitura	Mandatório
EtherHistoryBroadcastPkts	Contador	Leitura	Mandatório
EtherHistoryMulticastPkts	Contador	Leitura	Mandatório
EtherHistoryCRCAlignErrors	Contador	Leitura	Mandatório
EtherHistoryUndersizePkts	Contador	Leitura	Mandatório

EtherHistoryOversizePkts	Contador	Leitura	Mandatório
EtherHistoryFragments	Contador	Leitura	Mandatório
EtherHistoryCollisions	Contador	Leitura	Mandatório
EtherHistoryUtilization	Inteiro	Leitura	Mandatório

Tabela 4: Variáveis do Sistema.

O administrador pode incluir ou excluir variáveis no sistema, conforme seu conhecimento e experiência. Lembrando que esta liberdade na questão das variáveis se limita apenas às listas de variáveis disponíveis na definição da RMON-MIB 1, que faz parte do anexo 1.

7.4 Regras de Produção

Estas regras estão associadas a algumas variáveis da RMON-MIB, que interagem com a camada um e dois do modelo OSI. Atualmente já existe a RMON-MIB II que interage com as camadas de rede, transporte e aplicação, oferecendo possibilidades muito maiores de gerência de tráfego. Porém para a proposta aqui apresentada esta escolha não é relevante, já que o próprio administrador poderá escolher as variáveis que melhor lhe convierem.

Foram utilizadas para este modelo as mesmas nove regras do sistema Olho Vivo [SAENZ 96]. A quantidade e qualidade destas regras, pouco importam para o propósito deste trabalho. A única importância é que sirvam como modelo a fim de que o administrador possa entender como criar uma regra e como inseri-la no Sistema Especialista.

Diferentemente do Olho Vivo, o intuito deste trabalho é indicar o elemento de rede que possivelmente está gerando problema, e não sugestionar ações de como corrigir o problema em si, que é uma tarefa que deve ser do conhecimento de um administrador de redes.
Abaixo temos a tabela 5 onde são estabelecidas as regras de produção:

SE	Possível causa do problema
EtherHistoryCRCAlignErrors > 2% Ou EtherHistoryFragments > 2% Ou EtherHistoryJabbersPkts > 0% E EtherHistoryOversizePkts = 0% Ou EtherHistoryCollisions > 1%	Cabo
EtherHistoryCRCAlignErrors > 2%	Cabo AUI
EtherHistoryCRCAlignErrors > 2%	Conectores
EtherHistoryFragments > 2% Ou EtherHistoryCRCAlignErrors > 2% Ou EtherHistoryUndersizePkts > 2% Ou EtherHistoryOversizePkts > 2% Ou EtherHistoryJabbersPkts = EtherHistoryOversizePkts E EtherHistoryCollisions > 2%	Controladora
EtherHistoryFragments > 2% Ou EtherHistoryBroadcastPkts > 8% Ou EtherHistoryMulticastPkts > 8%	Host
EtherHistoryDropEvents > 1%	Memória
EtherHistoryFragments > 2%	Hub
EtherHistoryMulticastPkts > 8% Ou	Roteador

EtherHistoryFragments > 2%	
EtherHistoryCRCAlignErrors > 2% Ou EtherHistoryJabbersPkts = EtherHistoryOversizePkts E EtherHistoryCollisions < 2% Ou EtherHistoryFragments > 2% Ou EtherHistoryUndersizePkts > 2% Ou EtherHistoryOversizePkts > 2%	Transceiver

Tabela 5: Regras do Sistema.

Quanto mais regras possuir um sistema especialista, mais perto de uma resposta que seja a real causa do problema o sistema especialista chega. Por esta razão cabe ao administrador parcela de responsabilidade pelo sucesso desta ferramenta, pois é ele, que através de sua experiência e conhecimento da rede a qual administra, que pode incluir ou excluir novas variáveis e regras no sistema.

7.5. Sistema Especialista Proposto

O sistema especialista foi desenvolvido em Prolog [SWI], com o intuito de ser uma "Shell", onde o administrador pode utilizar o sistema com as regras existentes, e também tem a possibilidade, e esta é a intenção deste trabalho, de inserir as suas próprias regras através da interface com o usuário. Abaixo temos um esquema em blocos:

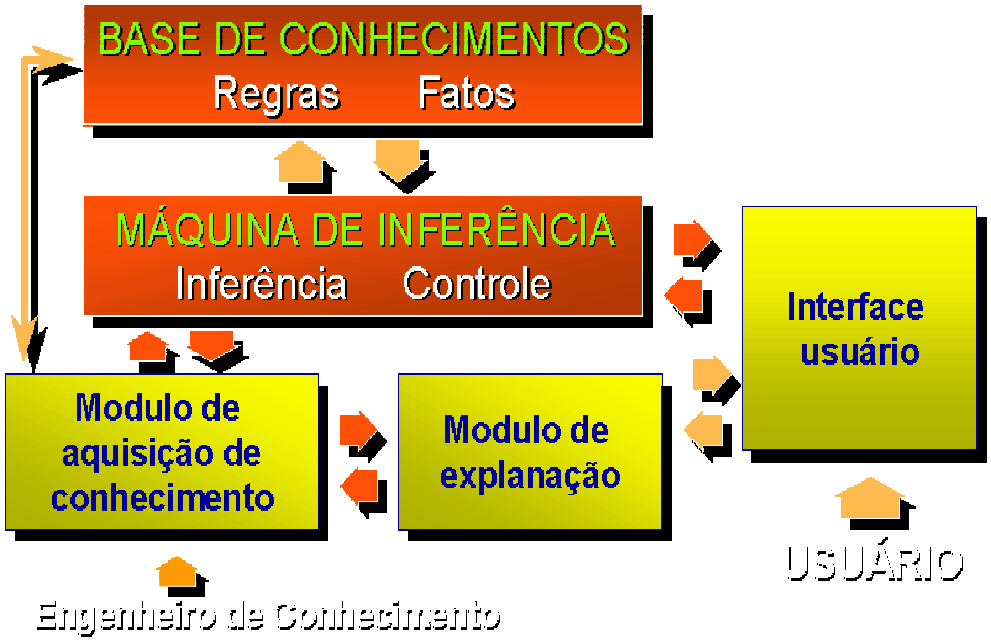


Figura 13: Detalhe em Blocos do Sistema Especialista

Foram cadastradas nove regras na base de conhecimento para demonstração. O sistema foi desenvolvido afim de que o administrador no momento que utilize esta ferramenta, seja estimulado a cadastrar novas regras que julgar importantes para as características da rede a qual gerencia. Nenhum conhecimento de sistema especialista ou de linguagem de programação é exigido.

Neste trabalho não foi tratado do módulo de explanação, ficando como sugestão para trabalhos futuros, este módulo se encarregaria, uma vez o sistema tendo chegado a uma conclusão do provável problema, de explicar o ***porque*** desta conclusão e principalmente ***como*** o sistema chegou a esta conclusão.

No módulo de interface com o usuário foi desenvolvido o seguinte menu:

SISTEMA MONITOR DE REDES

1. **INCLUIR REGRAS**
2. **EXCLUIR REGRAS**
3. **LISTAR REGRAS**
4. **EDITAR REGRAS**
5. **LISTAR FATOS**
6. **PARAMETRIZAR LEITURA DA MIB**
7. **RELER REGRAS**
8. **EXECUTAR LEITURA DA MIB/ RELER FATOS**
9. **IDENTIFICAR PROBLEMA**

S/s. SAIR

Figura 14: Detalhe do Menu do Sistema Especialista.

As opções 1, 2, 3, 4 e 7 referem-se às regras, nove regras foram cadastradas para demonstração de funcionamento do sistema, porém é possível ao administrador desenvolver as suas próprias regras, como também excluir as existentes. Há a opção também de listar todas as regras a fim de visualizar o conjunto inteiro.

A opção 5 mostra os fatos e seus valores.

A opção 6 fornece condições para que o administrador possa incluir ou excluir, de acordo com os grupos de variáveis disponíveis na RMON-MIB, as variáveis que considerar importantes para a monitoração de sua rede.

A opção 7 faz com que possíveis alterações das regras durante a execução do programa sejam relidas, atualizando a memória de trabalho.

A opção 8 dispara a interface SNMP para que uma leitura da base MIB seja efetuada imediatamente, atualizando os valores das variáveis no arquivo de fatos e também faz com que o Prolog releia este arquivo atualizando os fatos no sistema.

A opção 9 inicia a busca pelo possível problema, ou seja, a inferência sobre a base de conhecimento de posse dos fatos. Quando uma regra é disparada é enviada para a tela do computador uma mensagem informando o provável elemento da rede com problema.

As operações que interferem com a MIB e com as variáveis, estão sendo feitas por intermédio de chamadas ao programa desenvolvido em linguagem C, que é executado pela chamada através da interface em Prolog.

7.6. Simulações

Foram efetuadas as primeiras simulações com a interface SNMP a fim de validar o perfeito funcionamento das opções de menu propostas no programa conforme mostrado na figura 15:

Figura 15: Menu do programa Interface SNMP.

Na primeira opção de menu é efetuada uma consulta a variável, por exemplo, sysUpTime, a qual retorna a hora do sistema, conforme figura 14:

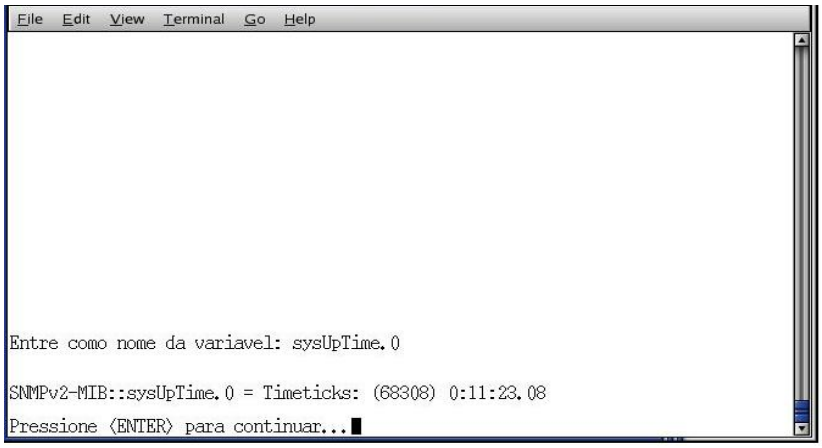


Figura 16: Consulta valor de variável na interface SNMP.

A próxima simulação é a inclusão da variável snmpProxyDrops no arquivo texto "fatos" a ser gerado para que o Sistema Especialista possa ler, tendo como condição que a mesma seja prevista pela versão da MIB corrente, conforme mostra a figura 15:

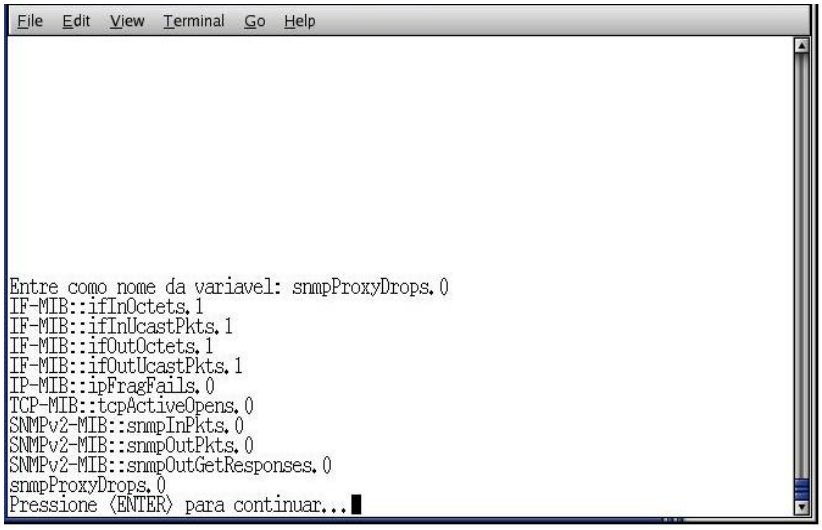


Figura 17: Inclusão de variável a ser monitorada.

A opção três do menu apresenta a possibilidade de remoção de uma variável do arquivo texto “fatos”, para a simulação foi retirada a mesma variável inserida acima, conforme Figura 16.

Figura 18: Remoção de variável da interface SNMP.

A opção quatro do menu possibilita a listagem das variáveis pré-selecionadas e seus valores, estas são a variáveis definidas para demonstração do sistema conforme figura 17:

Figura 19: Listagem de todas as variáveis e seus valores.

A última opção de menu, cinco, não gera nenhuma interface, simplesmente faz com que a geração ou atualização do arquivo texto com as variáveis e seus valores ocorra imediatamente.

Para a simulação do monitor foi gerado um arquivo de fatos, atribuindo valor elevado para a variável "Dropevents", como mostra a figura 18, para que ocorresse o disparo da regra relacionada à memória armazenada no arquivo "regras_rmon.txt", conforme figura 20, que representa a base de conhecimento inicial do sistema.

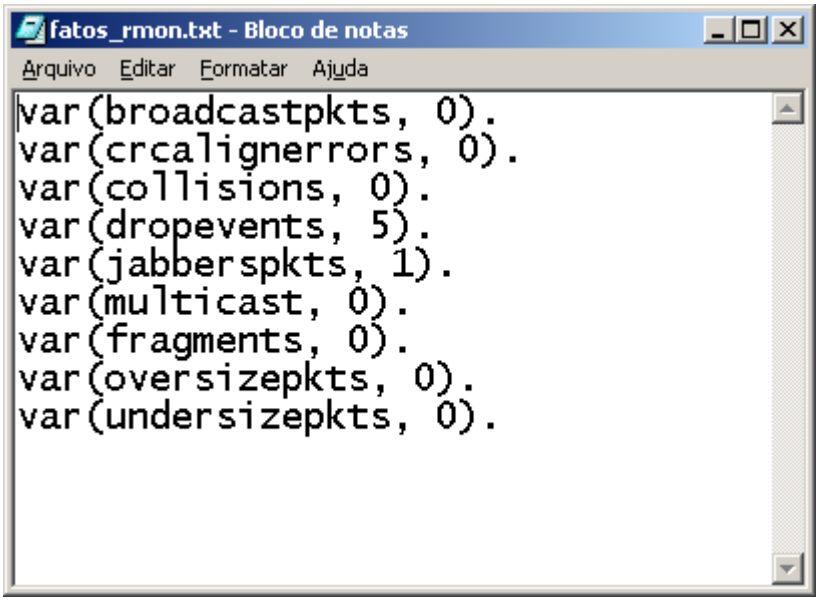


Figura 20: Arquivo fatos_rmon.txt para simulação.

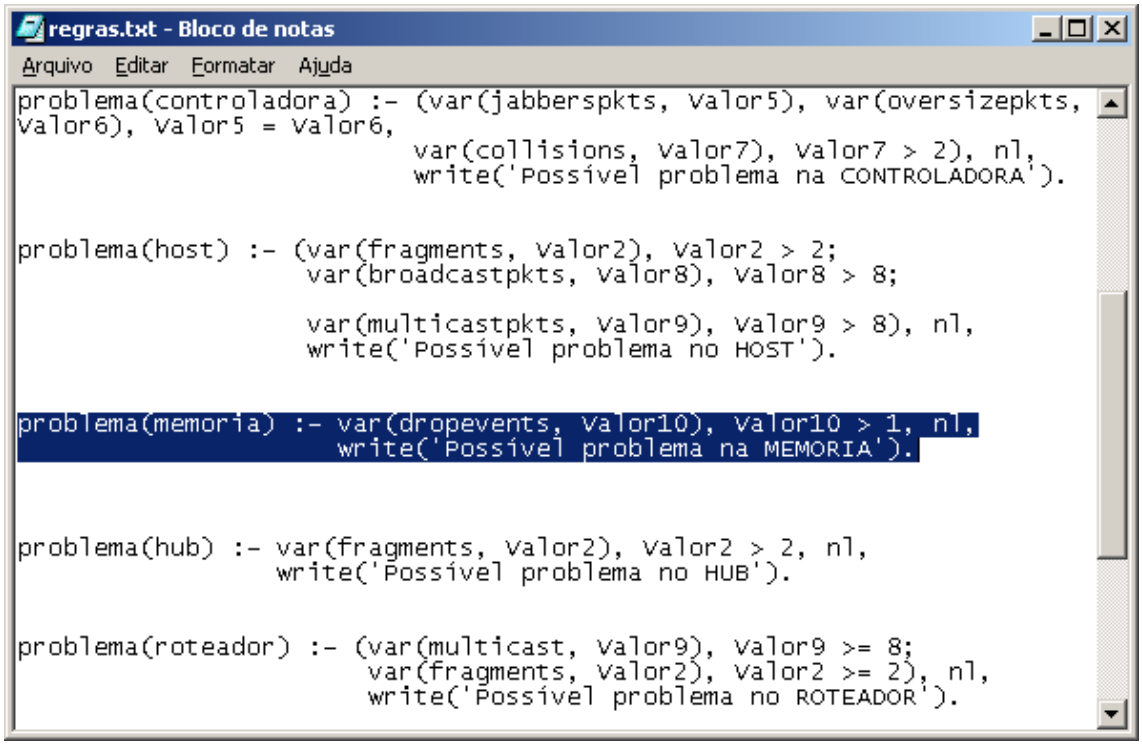


Figura 21: Arquivo de regras (base de conhecimento) do Monitor.

Em diagrama de blocos pode-se visualizar o funcionamento do programa monitor conforme a figura 20 abaixo:

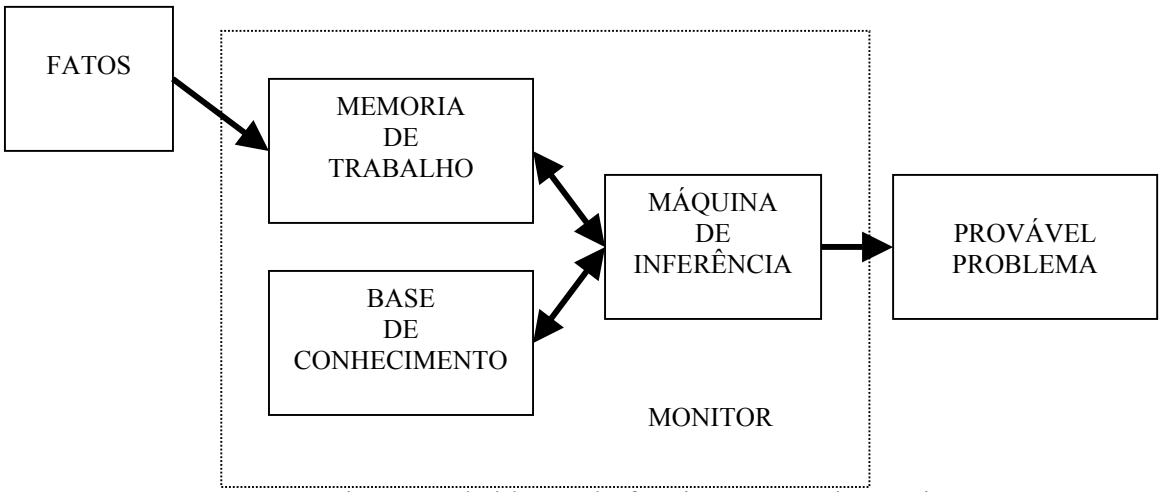


Figura 22: Diagrama de blocos do funcionamento do monitor.

Inicialmente o arquivo de fatos é atualizado automaticamente pelo monitor, quando a opção “identifica problema” é utilizada. Estas informações são mantidas na memória de

trabalho até uma próxima atualização. Neste momento, a máquina de inferência tenta encontrar na base de conhecimento alguma regra que possa ser disparada com a combinação dos fatos corrente na memória de trabalho. No caso do monitor os fatos são compostos pelo nome e valor da variável lida. Avaliando a figura 18 percebe-se que o valor da variável "dropevents" está com o valor excessivo de 5, representando alguma anormalidade na rede, isto acarretará o disparo da regra problema (memória), como pode ser visto pela figura 19. Uma vez não encontrando mais regras que possam ser disparadas, a máquina de inferência conclui que provavelmente há um problema com a memória do computador onde se encontra o agente e informa isto na tela para o usuário, como pode ser visto na figura 23.

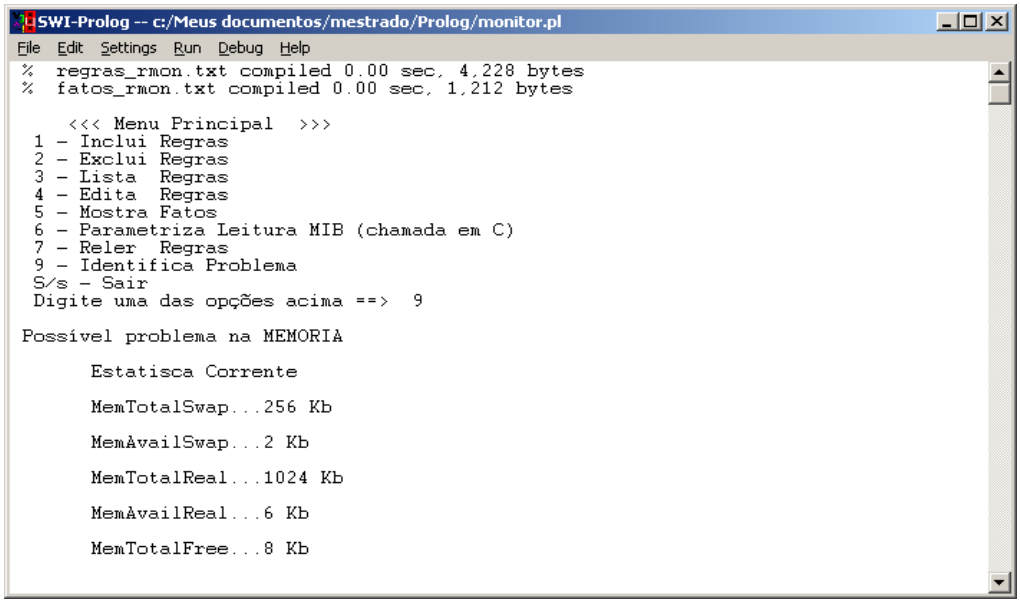


Figura 23: Simulação do monitor.

Neste capítulo foram demonstradas as dinâmicas de funcionamentos das implementações, a maior dificuldade foi em relação à interface SNMP, pela falta de ferramentas de código aberto, de preferência escritas em C, que se pudesse interagir. As versões das MIB's também são outro empecilho, pois além da dificuldade de se encontrar uma plataforma aberta, com seus códigos fontes, nem sempre estas plataformas contemplam todas as versões de MIB's existentes, por esta razão não utilizamos para a simulação uma RMON-MIB, mas sim uma MIB versão I, mas isto de forma nenhuma alterou ou desmereceu o resultado obtido.

Em relação à implementação em Prolog, é interminável, o poder da linguagem com o crescente conhecimento da mesma, possibilitam sempre que uma nova função seja implementada ou melhorada. E neste caso apenas o conhecimento e a imaginação são os limites.

CAPÍTULO 8 - CONCLUSÃO

A utilização da Inteligência Artificial e do protocolo SNMP para o monitoramento de redes de computadores, a fim de auxiliar o administrador na gerência, se mostrou adequada para esta aplicação. A proposta inicial foi comprovada nas simulações, variáveis foram monitoradas e mensagens foram emitidas, quando seus valores assumiram valores acima dos limites estabelecidos como normais.

8.1. Resumo do Trabalho

Através da Inteligência Artificial foi representado o conhecimento do administrador na monitoração de redes de computadores em forma de uma base de conhecimento. Através do protocolo SNMP foi possível a leitura dos parâmetros de rede. Nesta dissertação além do desenvolvimento teórico do monitor foi também implementado um Sistema Especialista na forma de um protótipo. Com a implementação prática do protótipo comparam-se os valores dos parâmetros de rede lidos da MIB com as regras de conhecimentos cadastradas no Sistema Especialista, quando algum valor se apresenta acima dos limites estabelecidos, um alerta é emitido.

8.2. Principais Resultados

O programa desenvolvido a fim de ler os objetos da MIB e fazer a integração com o Sistema Especialista funcionou perfeitamente. Este item foi de grande dificuldade, visto não existirem agentes com plataformas abertas e desenvolvidos em linguagem C para fácil integração. Com a pequena base de conhecimento cadastrada foi possível verificar o disparo de algumas regras em função dos valores dos parâmetros de rede que estavam acima do valor normal.

As vantagens de utilizar o Sistema Monitor podem ser citadas através de alguns pontos positivos observados durante o desenvolvimento da dissertação:

- Há possibilidade de selecionar os objetos de rede a serem monitorados;
- O administrador pode cadastrar suas próprias regras;
- Fácil adaptar para funcionar com várias versões de MIB's;
- O monitor se adequa às reais necessidades de controle.

8.4. Metodologia Adotada

No sistema implementado foi desenvolvido o software, onde podemos citar como principais atividades para se alcançar os objetivos desta dissertação:

- Identificação do Problema: Esta talvez tenha sido a fase mais complexa, pois há uma grande variedade de ferramentas de gerência de redes. Pela própria experiência e por várias entrevistas com administradores de redes, foi possível identificar uma linha de dissertação direcionada ao desenvolvimento de uma ferramenta que possibilitasse ao administrador interagir, adaptando esta ferramenta às suas necessidades, o que a maior parte das ferramentas existentes não permite;
- Desenvolvimento dos softwares: Foi necessário o desenvolvimento de dois programas, um desenvolvido em linguagem C para a leitura dos parâmetros de rede, outra para armazenar o conhecimento do administrador e poder disparar os alertas, desenvolvido em Prolog;
- Simulações e Ajustes: Foram executadas várias simulações lendo parâmetros de rede em situações críticas, observando o disparo das regras cadastradas. Vários ajustes foram feitos no Sistema Especialista a fim de possibilitar o fácil cadastro de novas regras pelo administrador, e a fim de apresentar os alertas com um conjunto de dados que auxiliasse na conclusão do problema.

De forma conclusiva a contribuição deste trabalho está no incentivo a participação do administrador de rede na adaptação da ferramenta às suas necessidades, já que possibilita que o administrador possa cadastrar seu conhecimento na base, o que lhe confere uma característica de escalabilidade, que permite que a ferramenta se torne mais robusta e útil com o tempo, liberando desta maneira o tempo dedicado à monitoração da

rede por parte do administrador, liberando-o para outras atividades necessárias à gerência da rede.

REFERÊNCIAS BIBLIOGRÁFICAS

[**BRACHMAN 88**] BRACHMAN, R.J., "The Basics of Knowledge Representation and Reasoning", AT&T Technical Journal, Vol.67, N.1, p. 15, 1988.

[BUCHANAN84] Buchanan, B. G., and E. H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments oJ the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley, 1984.

[CARVALHO97] Tereza Cristina Carvalho. RMON permite gerenciar redes remotas. LANTIMES Brasil, vol. 3, edição 1, pp. 24-25, Abril 1997.

[CASE90] J. Case, M. Fedor, M. Schoffstall e J. Davin. A Simple Network Management Protocol (SNMP). RFC 1157. Network Working Group. Maio 1990.

[CISCO96] Cisco Systems Inc. Simple Network Management Protocol (SNMP). In Cisco Systems Inc., Cisco Connection Documentation - Enterprises Series, San Jose, 1996.

[CISCOWORKS] CiscoWorks 2000 – Disponível on-line. http://www.cisco.com/warp/public/44/jump/ciscoworks.shtml

[FEIGENBAUM79] - Feigenbaum, M. J. [1979] "The universal metric properties of nonlinear transformations," *J. Stat. Phys.* **21**, 669-706.

[GIORGI96] Ulisses Ponticelli Giorgi. Tutorial Proxy. [online] Disponível na Internet via WWW. URL: http://penta.ufrgs.br/redes296/proxy/proxy.html 1996.

[HOLANDA 98] HOLANDA**, Raimir Holanda Filho.** Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores. Universidade Federal do Ceará. 1998.

[HARDAKER] HARDAKER, Wes. Ucd-snmp. www.sourceforge.net

[HEDRICK88] Charles L. Hedrick. Introduction to the Internet Protocols. Computer Science Facilities Group. Laboratory for Computer Science Research - Center for Computers and Information Services, The State University of New Jersey. 1988.

[LINDSAY80] Lindsay, R. K., B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg. *Application of Artificial Intelligence for Chemistry: The DENDRAL Project.* New York: McGraw-Hill, 1980.

[MICROSOFT] Visual Studio.Net, Visual C++ 6.0.
http://www.msdnbrasil.com.br/tecnologias/vsnet/default.aspx

[NETSCOUT96] NetScout Systems Inc. The Integration of RMON and SNMP Technologies for Managing Enterprise Networks. NetScout Documentation. 1996.

[ODA94a] Cybelle Suemi Oda. Introdução à Gerência de Redes. [online] Disponível na Internet via WWW. URL: http://www.gt-er.cg.org.br/operacoes/gerencia-redes 1994.

[ODA94b] Cybelle Suemi Oda. Gerenciamento de Redes de Computadores - Noções Básicas. [online] Disponível na Internet via WWW. URL: http://www.gt-er.cg.org.br/operacoes/gerencia-redes 1994.

[OPENVIEW] - HP Open View – Disponível on-line.
http://www.openview.hp.com/

[OTTEN 93] OTTEN, Hans. **The Beholder Cookbook**. Delft: Delft University, Netherlands, Dec.1993.

[**PEPER 91**] PEPER, G., "Hypertext: Its Relationship to, and Potencial Impact on, Knowledge Based Systems", 1991.

[PROCESS97] Process Software Corporation. Web Proxy Servers. In Purveyors Administrator's Guide. [online] Disponível na Internet via WWW. URL: http://vms.process.com/~help/help.html 1997.

[ROSE90] M. Rose, K. McCloghrie. Management Information Base for Network Management of TCP/IP-based internets. RFC 1156. Network Working Group. 1990.

[SAENZ 96] SÁENZ A. Esmilda. **Olho Vivo Sistema Especialista para Gerência Pró-ativa Remota.** Porto Alegre: CPGCC da UFRGS, 1996.

[SOARES97] Luiz Fernando Gomes Soares, Guido Lemos de Souza Filho e Sérgio Colcher. Redes de computadores: das LANs, MANs e WANs às redes ATM. Campus, 2º edição, 1997.

[SWI] SWI-Prolog. http://www.swi-prolog.org/

[TAROUCO93] Liane Margarida Rockenbach Tarouco. Evolução do gerenciamento de Redes. In Sociedade Brasileira para Interconexão de Sistemas Abertos, Ed., Gerenciamento de Redes - Uma abordagem de sistemas abertos, pp. 1-12. Makron Books do Brasil, São Paulo, 1993.

[TIVOLI] – Tivoli Enterprise – Dispnível on-line.
http:// www.ibm.com/software/tivoli/

[WALDBUSSER91] S. Waldbusser. Remote Network Monitoring Management Information Base. RFC 1271. Carnegie Mellon University. 1991.

ANEXO

Definição das principais variáveis da RMON-MIB:

```
EtherStatsEntry ::= SEQUENCE {
   etherStatsIndex                Integer32,
   etherStatsDataSource           OBJECT IDENTIFIER,
```

```
    etherStatsDropEvents            Counter32,
    etherStatsOctets                Counter32,
    etherStatsPkts                  Counter32,
    etherStatsBroadcastPkts         Counter32,
    etherStatsMulticastPkts         Counter32,
    etherStatsCRCAlignErrors        Counter32,
    etherStatsUndersizePkts         Counter32,
    etherStatsOversizePkts          Counter32,
    etherStatsFragments             Counter32,
    etherStatsJabbers               Counter32,
    etherStatsCollisions            Counter32,
    etherStatsPkts64Octets          Counter32,
    etherStatsPkts65to127Octets     Counter32,
    etherStatsPkts128to255Octets    Counter32,
    etherStatsPkts256to511Octets    Counter32,
    etherStatsPkts512to1023Octets   Counter32,
    etherStatsPkts1024to1518Octets  Counter32,
    etherStatsOwner                 OwnerString,
    etherStatsStatus                EntryStatus
}

EtherHistoryEntry ::= SEQUENCE {
    etherHistoryIndex               Integer32,
    etherHistorySampleIndex         Integer32,
    etherHistoryIntervalStart       TimeTicks,
    etherHistoryDropEvents          Counter32,
    etherHistoryOctets              Counter32,
    etherHistoryPkts                Counter32,
    etherHistoryBroadcastPkts       Counter32,
    etherHistoryMulticastPkts       Counter32,
    etherHistoryCRCAlignErrors      Counter32,
    etherHistoryUndersizePkts       Counter32,
    etherHistoryOversizePkts        Counter32,
    etherHistoryFragments           Counter32,
    etherHistoryJabbers             Counter32,
    etherHistoryCollisions          Counter32,
    etherHistoryUtilization         Integer32
}

AlarmEntry ::= SEQUENCE {
    alarmIndex                  Integer32,
    alarmInterval               Integer32,
    alarmVariable               OBJECT IDENTIFIER,
    alarmSampleType             INTEGER,
    alarmValue                  Integer32,
```

```
   alarmStartupAlarm            INTEGER,
   alarmRisingThreshold         Integer32,
   alarmFallingThreshold        Integer32,
   alarmRisingEventIndex        Integer32,
   alarmFallingEventIndex       Integer32,
   alarmOwner                   OwnerString,
   alarmStatus                  EntryStatus
}

HostEntry ::= SEQUENCE {
   hostAddress            OCTET STRING,
   hostCreationOrder      Integer32,
   hostIndex              Integer32,
   hostInPkts             Counter32,
   hostOutPkts            Counter32,
   hostInOctets           Counter32,
   hostOutOctets          Counter32,
   hostOutErrors          Counter32,
   hostOutBroadcastPkts   Counter32,
   hostOutMulticastPkts   Counter32
}

HostTimeEntry ::= SEQUENCE {
   hostTimeAddress            OCTET STRING,
   hostTimeCreationOrder      Integer32,
   hostTimeIndex              Integer32,
   hostTimeInPkts             Counter32,
   hostTimeOutPkts            Counter32,
   hostTimeInOctets           Counter32,
   hostTimeOutOctets          Counter32,
   hostTimeOutErrors          Counter32,
   hostTimeOutBroadcastPkts   Counter32,
   hostTimeOutMulticastPkts   Counter32
}
```

Programa da interface SNMP:

```
#include "stdio.h"
#include "stdlib.h"
#include "ctype.h"

int             failures = 0;

#define NETSNMP_DS_APP_DONT_FIX_PDUS 0
```

```
void menu();
void clearscreen();
void readvar();
int readvars(char *var, char* result);
void showresult(int line, char* result);
void addvar();
void removevar();
void listvar();
void generatefile();
void processvar(char *var, FILE *fout);
void pause();
int readfile(char* filename, char* result);
void writefile(char* filename, char *data);

int main(int argc, char *argv[]) {
	if (argc > 1) {
		if (strcmp(argv[1], "-generatefile") == 0) {
			generatefile();
			exit(0);
		}
	}

	menu();

	clearscreen();
}

void menu() {
	char op = '1';
	while (op != '0') {
		clearscreen();
		printf("	Escolha o que voce quer fazer\n");
		printf("	1-> Consultar uma variavel\n");
		printf("	2-> Adicionar uma variavel\n");
		printf("	3-> Remover uma variavel\n");
		printf("	4-> Listar variaveis programadas\n");
		printf("	5-> Gerar o arquivo\n");
		printf("	0-> Sair\n");
		printf("	--> ");

		op = getchar();
		switch(op) {
			case '1':
				readvar();
				break;
			case '2':
				addvar();
				break;
```

```
                        case '3':
                                removevar();
                                break;
                        case '4':
                                listvar();
                                break;
                        case '5':
                                generatefile();
                                break;
                }
        }
}

void clearscreen() {
        printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
}

void readvar() {
        clearscreen();
        printf("Entre como nome da variavel: ");
        char line[256];
        char result[10000];
        scanf("%s", line);
        int lines = readvars(line, result);
        showresult(lines, result);
}

void showresult(int line, char* result) {
        clearscreen();
        printf("\n\n%s\n\n", result);
        pause();
}

void addvar() {
        clearscreen();
        printf("Entre como nome da variavel: ");
        char line[256];
        char result[10000];
        scanf("%s", line);

        int lines = readfile("/var/monitor.vars", result);
        if (strstr(result, line) == 0) {
                strcat(result, "\n");
                strcat(result, line);
                printf("%s\n", result);
                writefile("/var/monitor.vars", result);
        }
        pause();
```

```
}

void removevar() {
	clearscreen();
	printf("Entre como nome da variavel: ");
	char line[256];
	char result[10000];
	char source[10000];
	scanf("%s", line);

	int lines = readfile("/var/monitor.vars", source);
	if (strstr(source, line) != 0) {
		char var[256];
		char ch;
		int pos = 0;
		int varpos = 0;
		result[0] = 0;
		ch = source[pos];
		do {
			if (ch == '\n' || ch == 0) {
				if (strcmp(var, line) != 0) {
					strcat(result, var);
					if (ch == '\n') {
						strcat(result, "\n");
					}
				}
				varpos = 0;
			} else {
				var[varpos] = ch;
				var[++varpos] = 0;
			}
			pos++;
			if (ch == 0) {
				break;
			}
			ch = source[pos];
		} while (1);
		writefile("/var/monitor.vars", result);
	}
	printf("%s\n", result);
	pause();
}

void listvar() {
	char result[10000];
	int lines;

	lines = readfile("/var/monitor.vars", result);
```

```
	printf("%s\n", result);
	pause();
}

void generatefile() {
	char values[10000];
	char line[10000];
	int lines = readfile("/var/monitor.vars", line);
	char ch;
	int i = 0;
	for (i = 0; i < strlen(line); i++) {
		if (line[i] == '\n') {
			line[i] = ' ';
		}
	}

	lines = readvars(line, values);

	FILE *fout;
	fout = fopen("/var/monitor.log", "w");

	char var[256];
	int pos1 = 0;
	int pos2 = 0;
	do {
		char ch = values[pos1];
		if (ch == '\n' || ch == 0) {
			processvar(var, fout);
			var[0] = 0;
			pos2 = 0;
		} else {
			var[pos2] = ch;
			var[++pos2] = 0;
		}

		if (ch == 0) {
			break;
		}
		pos1++;
	} while (1);
	fclose(fout);
}

void processvar(char *var, FILE *fout) {
	int i = 0;
	int action = 0;
	char varname[128];
	char value[256];
```

```
        int posvar = 0;
        int posvalue = 0;
        for (i = 0; i < strlen(var); i++) {
                char ch = var[i];
                switch(action) {
                        case 0:
                                if (ch == ':') {
                                        i++;
                                        action = 1;
                                }
                                break;
                        case 1:
                                if (ch == ' ') {
                                        action = 2;
                                        i++;
                                        i++;
                                        i++;
                                        posvalue = 0;
                                } else {
                                        varname[posvar] = ch;
                                        varname[++posvar] = 0;
                                }
                                break;
                        case 2:
                                if (ch == ':') {
                                        posvalue = 0;
                                } else {
                                        value[posvalue] = ch;
                                        value[++posvalue] = 0;
                                }
                                break;
                }
        }
        char line[256];
        strcpy(line, "var (");
        strcat(line, varname);
        strcat(line, ",");
        strcat(line, value);
        strcat(line, ")\n");
        fprintf(fout, "%s", line);
        fflush(fout);
}

int readvars(char *var, char *result) {
        char cmd[5000];
        int lines = 0;
        strcpy(cmd, "snmpget localhost ");
        strcat(cmd, var);
```

```
	strcat(cmd, " >/var/monitor.tmp");
	system(cmd);
	lines = readfile("/var/monitor.tmp", result);
	remove("/var/monitor.tmp");
	return lines;
}

int readfile(char* filename, char* result) {
	int lines = 0;
	char line[256];
	char last_line[256];
	FILE *fp;
	last_line[0] = 0;
	fp = fopen(filename, "r");
	while (!feof(fp)) {
		fgets(line, 255, fp);
		if (strlen(line) > 2 && strcmp(line, last_line) != 0) {
			if (lines == 0) {
				strcpy(result, line);
			} else {
				strcat(result, line);
			}
			lines++;
			strcpy(last_line, line);
		}
	}
	fclose(fp);
	if (result[strlen(result) - 1] == '\n') {
		result[strlen(result) - 1] = 0;
	}
}

void writefile(char *filename, char *data) {
	FILE *fp;
	fp = fopen(filename, "w");
	fprintf(fp, "%s", data);
	fclose(fp);
}

void pause() {
	printf("Pressione <ENTER> para continuar...");
	getchar();
	getchar();
}
```